Canonical Solution Groups of a Homomorphism

Manipulator Kinematics, Nonparametric Regression and Distributed Object Systems

Conor Jeremiah Doherty

Doctor Of Philosophy

The University of Aston in Birmingham

July 1997

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Canonical Solution Groups of a Homomorphism

Manipulator Kinematics, Nonparametric Regression and Distributed Object Systems

CONOR J DOHERTY

Doctor Of Philosophy, 1997

Thesis Summary

The kinematic mapping of a rigid open-link manipulator is a homomorphism between Lie groups. The homomorphism has solution groups that act on an inverse kinematic solution element. A canonical representation of solution group operators that act on a solution element of three and seven degree-of-freedom (*dof*) dextrous manipulators is determined by geometric analysis. Seven canonical solution groups are determined for the seven *dof* Robotics Research K-1207 and Hollerbach arms. The solution element of a dextrous manipulator is a collection of trivial fibre bundles with solution fibres homotopic to the Torus. If fibre solutions are parameterised by a scalar, a direct inverse function that maps the scalar and Cartesian base space coordinates to solution element fibre coordinates may be defined. A direct inverse parameterisation of a solution element may be approximated by a local linear map generated by an inverse augmented Jacobian correction of a linear interpolation. The action of canonical solution group operators on a local linear approximation of the solution element of inverse kinematics of dextrous manipulators generates cyclical solutions. The solution representation is proposed as a model of inverse kinematic transformations in primate nervous systems.

Simultaneous calibration of a composition of stereo-camera and manipulator kinematic models is under-determined by equi-output parameter groups in the composition of stereo-camera and Denavit Hartenberg (DH) models. An error measure for simultaneous calibration of a composition of models is derived and parameter subsets with no equi-output groups are determined by numerical experiments to simultaneously calibrate the composition of homogeneous or pan-tilt stereo-camera with DH models.

For acceleration of exact Newton second-order re-calibration of DH parameters after a sequential calibration of stereo-camera and DH parameters, an optimal numerical evaluation of DH matrix first order and second order error derivatives with respect to a re-calibration error function is derived, implemented and tested.

A distributed object environment for point and click image-based tele-command of manipulators and stereo-cameras is specified and implemented that supports rapid prototyping of numerical experiments in distributed system control. The environment is validated by a hierarchical k-fold cross validated calibration to Cartesian space of a radial basis function regression correction of an affine stereo model.

Basic design and performance requirements are defined for scalable virtual micro-kernels that broker inter-Java-virtual-machine remote method invocations between components of secure manageable fault-tolerant open distributed agile Total Quality Managed ISO 9000+ conformant Just in Time manufacturing systems.

Keywords: Solution Groups, Manipulator Inverse Kinematics, Model Calibration, Nonparametric Regression, Distributed Object Software Engineering, Sensorimotor Coordination

And pluck till time and times are done
The silver apples of the moon,
The golden apples of the sun.

W.B. Yeats. *The Song of Wandering Aengus*

To Doreen

Acknowledgements

I would like to express my deepest gratitude to my supervisor Richard Rohwer for his guidance and understanding of this thesis. My ideas were considerably clarified and amplified by his invaluable insights, suggestions, comments and corrections. He taught me how to marshal, deepen and sharpen my thoughts in order to define and analyse a problem thoroughly.

The risk of a research project that spans several disciplines is that it is difficult to attain the same depth in each discipline. I am grateful to the many people who assisted me to identify and deepen my understanding of the basic multi-disciplinary ideas of what I hope has become a coherent thesis. These include Christopher Williams, Christopher Bishop, David Lowe, Eric Amador, Man Kwan Ng, David Barber, Huaiyu Zhuh, Alan McLachlan, Ian Nabney, Michael Tipping, Michael Morciniec, Cazhaow Qazaz, Markus Svensén, Ansgar West, and Patrick van der Smagt.

Finally, I would like to thank numerous software engineers at Iona Technologies for helping me to understand the scaling limits of distributed object systems.

I gratefully acknowledge the financial support of a PhD studentship from the University of Aston in Birmingham.

Conor Doherty

Dublin
July 1997

Contents

List of Figures

List of Tables

List of Definitions

Definition

Symbols and Notation

In general blackboard bold letters, e.g. $\mathbb{N}$, denote sets. Fraktur letters, e.g. G, denote groups. Euler Fraktur letters, e.g. $\mathfrak{F}$, denote spaces. Upper-case bold letters, e.g. $\mathbf{A}$, denote matrices. Lower-case bold letters, e.g. $\mathbf{x}_i$ denote vectors. Calligraphical letters, e.g. $\mathcal{F}$, denote mappings and homomorphisms.

Sets

$\mathbb{CP}$	A set of critical points of a homomorphism.
$\mathbb{CPS}$	A critical point surface of a homomorphism.
$\mathbb{CR}$	A set of co-regular points of a homomorphism.
$\mathbb{D}$	A data set.
$\mathbb{M}$	A manifold.
$\mathbb{N}$	The set of Natural numbers.
$\mathbb{O}$	An orbit of a group G acting on L.
$\mathbb{R}$	The set of Real numbers.
$\mathbb{V}$	A Voronoi partition of a set.
$\mathbf{V}_{-i}$	A calibration data set.
$\mathbf{V}_i$	A validation data set.
m	A set of DH parameters.
$\aleph_{\mathbf{x}}$	A tesselation of $\mathbf{x}_i$, a solution element base subspace.
$\mathbb{T}$	A topographic map.
$\mathbb{S}$	A topographic regression map.

Groups

C	The Cartesian group.
C^n	The Cartesian group. If $n = 6$, $\mathsf{C}^6 = \mathsf{SE}(3)$.
C_n	The cyclic group of order n.
F	A group.
G	A group.
G_k	A solution group factor of G.
G_i^j	The solution group factors of G_i.
$\mathsf{G/H}$	A factor group.
GL	The general linear group.
H	A subgroup.
L	A Lie group, usually of the homomorphism $\mathcal{F}$.
$\mathsf{L/G}$	A quotient manifold.
O	The Orthogonal group.
$\mathsf{SE}(n)$	The Special Euclidean group.
S^n	The n-sphere group.
$\mathsf{SO}(n)$	The Special Orthogonal group.
T^n	The n-Torus group.
Q	The Quaternion group.

Vector Spaces

$\mathfrak{B}$	A base space.
$\mathfrak{F}$	A fibre.
$\mathfrak{FB}$	A fibre bundle.

$\mathfrak{W}$	A workspace. This is a base space of $\mathcal{F}$ fibre bundle.
$\mathfrak{W}_w$	A singularity free sub-workspace w of $\mathfrak{W}$.
Ω	A metric space.

Coordinate Systems

$\{A\}_i$	A coordinate frame.
$\mathfrak{C}$	A Cartesian coordinate system.
$\mathfrak{J}$	A joint coordinate system.
$\{X_i, Y_i, Z_i\}$	The orthogonal axes of a frame.
${}^a_b\mathbf{T}$	A homogenous transform.

Matrices

$\mathbf{A}$	An affine transform.
DH	A Denavit Hartenberg Matrix.
$\mathbf{J}$	A Jacobian.
$\mathbf{H}$	A Hessian.
$\mathbf{P}$	A perspective projection transform or a prismatic joint.
$\mathbf{R}$	A camera displacement, a rotation transform or a rotational joint.
$\mathbf{S}$	An affine stereo-camera model.
$\mathbf{T}_j$	A homogeneous transform j.
$\mathcal{M}_1$	Interior camera model.
$\mathcal{M}_2$	Exterior camera model.

Vectors

$\boldsymbol{\alpha}$	Denavit Hartenberg link twist parameters.
$\mathbf{a}$	Denavit Hartenberg link length parameters.
$\mathbf{c}_k$	Centres of radial basis functions.
$\mathbf{d}$	Denavit Hartenberg link distance parameters.
$\mathbf{G}$	A set of Cartesian vectors that define a wireframe model of a manipulator.
ζ	A parameterisation of a fibre.
η_t	A gradient descent step size at time t.
ϕ_k	A basis function, usually radial.
μ	A quaternion.
ν	An eigenvalue.
$\mathbf{o}_t$	System outputs.
$\boldsymbol{\sigma}_k$	Variances of radial basis functions.
$\boldsymbol{\theta}_i$	Joint coordinates.
$\mathbf{u}_t$	Control inputs.
$\mathbf{v}$	A vector field.
$\mathbf{x}_g$	A Cartesian goal position coordinates.
$\mathbf{x}_i$	Cartesian coordinates.
$\mathbf{x}^i$	$\mathbf{x}^i\mathbf{T}^i = \prod_{j=1}^{n} \mathbf{T}_j$ evaluated at $\boldsymbol{\theta}_i$.
${}_j\mathbf{x}_i$	A point i in camera j coordinate system.
${}_j\mathbf{x}'_i$	A point i in image coordinate system of camera j.
$\boldsymbol{\tau}_t$	A vector of forces acting on a manipulator.
$\mathbf{w}$	The parameters of a model.
$\mathbf{y}_i$	A solution group solution element. $\mathbf{y}_i = (\mathbf{x}_i, \boldsymbol{\theta}_i)$.
${}^r\mathbf{y}_i^s$	A solution group s solution element of L in representation r.
ϖ	A random variable or noise function.
ω	A pitch parameter in a fixed angle orientation convention.
γ	A roll parameter in a fixed angle orientation convention.
λ	A yaw parameter in a fixed angle orientation convention.

Scalars

f	The focal length of a lens.
t	Time.

w^{2*}	A regression regulariser hyper-parameter.
w^{1*}	A regression bias hyper-parameter.

Homomorphisms

$\mathcal{F}$	A forward kinematic mapping $\mathcal{F} : \boldsymbol{\theta}_i \in \mathfrak{J}^n \rightarrow \mathbf{x}_i \in \mathfrak{C}^m \equiv \mathsf{SE}(3) : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$
$\mathcal{F}^{-1}$	An inverse kinematic mapping $\mathcal{F}^{-1} : \mathfrak{C}^m \rightarrow \mathfrak{J}^n : \{\boldsymbol{\theta}_i\} = \mathcal{F}^{-1}(\mathbf{x}_i)$
F	An approximation of a forward kinematic function. $\mathbf{x}_i = F(\boldsymbol{\theta}_i)$ (F also refers to any non-parametric model).
F^{-1}	An approximation of an inverse kinematic mapping.
F_i	a set of non-parametric models.

Functions

f	A function.
β	Task functions to augment a Jacobian.
E^p	Manufacturing effeciency for product p.
F^p	Manufacturing flexibility for product p.
Γ_w	A canonical coordinatisation function of a solution fibre over a base space $\mathfrak{W}_w$.
L_1	A measure of representational expression compactness of ${}^r\Psi^s_k, {}^r\Phi^s_k$.
L_2	A measure of the compactness of representation ${}^r\mathbf{y}^s_i$.
L_3	A measure of the surface area of ${}^r\mathbf{y}^s_i$.
L	A Lyapunov function.
M	Manipulability.
ι	A Lie group inversion map.
ξ	A Lie group operation map.
π	Function for projection of a fibre over $\mathbf{x}_i$ to $\mathbf{x}_i$, $\mathbf{x}_i = \pi(\mathbf{x}_i, \boldsymbol{\theta}_i)$.
${}^r\rho^s_o$	A boundary surface of ${}^r\mathbf{y}^s_i$.
ϕ_α	Coordinate functions for a chart α.
U_α	A chart of a manifold.
A_r	An atlas $\{U^r_\alpha, \phi^r_\alpha\}$ of a representation r of L.
Υ^{-1}_w	A direct inverse function of a solution fibre over a base space $\mathfrak{W}_w$.

Binary Operators

$\otimes$	A group operator.
$\oplus$	A group operator.
$[,]$	The Lie bracket.
$[]$ $[]$	A norm.
$\|\|$ $\|\|$	The L_2 Euclidean norm.
$.*$	Element wise matrix multiplication following a Matlab® convention.

Unary Operators

Δ	An increment.
∇	A gradient.
$\|$ $\|$	A determinant.
dim	A dimension in $\mathbb{R}^n = n$.
ker	A kernel operator of a linear transformation. $\forall \mathbf{x} \neq 0, \mathbf{A}\mathbf{x} = 0 \iff \mathbf{x} \in \ker \mathbf{A}$.
rank	The number of linearly independent rows or columns of $\mathbf{A}$.
$\mathbf{J}^{-1}$	The inverse of a matrix $\mathbf{J}$.
$\mathbf{J}^{+}$	A pseudoinverse of a matrix $\mathbf{J}$.
$\mathbf{J}^{\Diamond}$	An augmented Jacobian.
${}^r\Psi^s_k$	A solution group s factor k in representation r that operates on $\boldsymbol{\theta}_i$ fibre of $\mathcal{F}$.
${}^r\Phi^s_k$	A solution group s factor k in representation r that operates on $\mathbf{x}_i \in \mathfrak{W}$ base space of $\mathcal{F}$.
$\exp(t\mathbf{v})\mathbf{y}_i$	The infinitesimal generator of the flow of a vector field $\mathbf{v}$ on L.

Distributed Object Systems

o_m	A computational object m.
${}^m_n\mathbf{e}^{(p,c,ex)}_t$	A communication event of priority p in context $\mathbf{c}$ from object o_m to o_n with processing

	exception vector ex at time t. ${}^{m}_{n}\mathbf{e}^{p}_{t} = \{{}^{m}_{n}i^{p}_{t}, {}^{m}_{n}s^{p}_{t}, {}^{m}_{n}n^{p}_{t}\}$.
${}^{m}_{n}i^{p}_{t}$	A synchronous invocation from o_m to o_n at time t.
${}^{m}_{n}s^{p}_{t}$	A stream from o_m to o_n at time t.
${}^{m}_{n}n^{p}_{t}$	A notification from o_m to o_n at time t.
v^{j}_{i}	A server j group of objects o_m attatched to kernel k_i.
k_i	An event routing micro-kernel.
$\mathtt{ref}_{(i,n)}$	A distributed pointer to object o_n attatched to kernel k_i.
Γ_s	A measure of a distributed system scale.
F^{p}_{s}	The probability of a fault at any system level.
D_s	The probable dependability of a system s.
R_s	The probable reliability of a system s.
A_s	The probable availability of a system s.
S_s	The probable safety of a system s.
M_s	The probable maintainability of a system s.
P_s	The probable performance of a system s.
P^{μ}_{s}	A measure of system performance.
F_s	The failure rate of a system s.
F^{a}_{s}	The burn in failure rate of a system s.
F^{b}_{s}	The stable operational failure rate of a system s.
F^{c}_{s}	The burn out failure rate of a system s.
C_s	The fault coverage of a system s.
$\bar{T}^{F}_{s}$	The mean time to failure of a system s.
$\bar{T}^{U}_{s}$	The mean time to recovery of a system s.
$\bar{T}_{s}^{(F^{1}_{s}, F^{2}_{s})}$	the mean time between failures F^{1}_{s}, F^{2}_{s} of a system s.
Ω	The distributed system design and engineering function.
ω_s	A distributed system code.
ϑ_s	Software developer resources $r^{1}_{s}, \ldots, r^{p}_{s}$.
χ_c	A software engineering methodology.
$\mho_s$	A requirements specification for a distributed system s.
$\bar{k}^{e}_{i}$	A kernel mean event processing duration.
$\bar{k}^{p}_{i}$	A kernel mean throughput.
$\bar{k}^{a}_{i}$	A kernel mean throughput absorption.
$\bar{k}^{u}_{i}$	A kernel utilsation rate.
$\bar{k}^{q}_{i}$	The mean length of the event queue on kernel i.
k^{w}_{i}	A kernel workload cycle time.
k^{r}_{i}	A kernel responsiveness time.
k^{c}_{i}	A kernel cycle time.
P_d	A network event transport protocol d.

Chapter 1

Introduction

Total Quality Management (TQM) (see (Saylor, 1991)) of ISO 9000+ conformant integrated Just in Time (JIT) (see (Harrison, 1992)) manufacturing systems requires continuous optimisation of business, social, environmental, design, engineering, logistic, operational, and process factors of production. Automation of the last five factors is a manufacturing priority in order to gain competitive advantage. The penetration of automation technologies into these factors varies according to industrial sector depending on the relative priorities of product innovation rate, efficiency of supply and distribution channels, plant productivity, total quality, process and assembly effeciency and flexibility.

Maximisation of integrated automation of the five mechanisable business factors makes possible concurrent design and engineering of short lifecycle rapid turnaround multi-feature customisable products. Tightly coupled manufacturing design, planning and operations erodes the classic Fordist (see (Batchelor, 1994)) separation of task conception, design and management from Taylorised division of production into sequential specialist atomic labour operations. Elements of the business intellectual function such as analysis and tactical decision making are devolved to the shop floor because of the complexity of computer controlled process organisation and reconfiguration. TQM JIT manufacturing systems require re-integration of intellectual and physical factory labour, especially at the systems maintenance and reconfiguration level.

The fixed costs of production escalate with technological investment and manufacturing output. Robotics and automation increase flexibility of production output in terms of product variation but reduce production output flexibly in terms of absolute magnitudes of products. Robotics and automation intensive production facilities are expensive. Expensive plants must operate at close to maximum capacity over 24 hours in order to recover fixed investment costs and eventually to generate a profit stream. With higher levels[1] of process automation, there is less managerial margin for maneuver at the labour supply regulation level both in terms of labour unit costs (relatively high for the residual non-automatable production components) and labour unit volumes contracted at a particular instant. Operational cost inflexibility complicate the cost effectivity analysis of return on investment of automation and robotics technologies.

An economic barrier to more widespread robotic and automation displacement of physical labour in addition to the low functional limits of robotic technology is that deployment is often only cost effective in high labour cost inverted population pyramid economies for production of high volume complex evolving artifacts with very high manufacturing tolerances, for instance vehicle assembly. In addition, the introduction of robotics and automation technologies tremendously complicates production system

[1] For instance, the current human to robot work force numerical ratio is on average about 0.2 in vehicle assembly

complexity, increasing overall system complexity, at the software and process level, by one to two orders of magnitude.

However, reduction of time to market of designs responding to market requirements drift achieved by tighter faster integration of design, prototyping and lean JIT production is competitive against classic scale inefficiencies of repetitive manual labour intensive mass market production organisation. If economic analysis indicates that investment in heavy automation is strategic for commercial survival, investment must be timed to coincide with international economic expansion cycles to neutralise short to medium term risks of investment write off caused by low demand.

Design of interlocking design and production processes is an important determinant of the success of automated assembly. Modular part design methodologies combined with logistic and assembly technology enable parts to be reused in different products. Products made from simple standard multi-product parts rather than complex parts that are unique to each product require more complex assembly but parts fabrication is less complex. Designing flexibility into parts and maximising assembly efficiency decouples flexibility from efficiency since manufacturing reconfiguration is simplified. In TQM ISO 9000 conformant JIT production systems, manufacture of variable products can be achieved more easily and economically during assembly than during high tolerance parts fabrication. Industrial robots are deployed to increase flexibility of automated process and assembly operations.

A robot manipulator is a programmable multi-functional mechanism usually deployed in assembly tasks. Many industrial manipulators in structured workspaces such as micro-electronic and car plants are programmed with *teach by show* (Craig, 1989). Teach by show programming is appropriate in standardised mass production environments where there is infrequent production reconfiguration or variation. Technicians indicate and record repetitive operation sequences of a manipulator arm, for instance gripping a paint sprayer, using teaching pendants. Correct sequences play back during assembly requires high manipulator repeatability, the ability to return to stored positions. Teach by showing does not require manipulators to have sensors to measure task space variables. New manipulator operation sequences must be recorded for each slightly different production task.

Advanced manufacturing replaces standardised mass-production with agile flexible manufacture of made-to-order small product lots. More advanced sensor and model-based manipulator programming optimises the advanced manufacturing tradeoff between efficiency and flexibility.

1.1 Definition (Manufacturing Efficiency)
Manufacturing efficiency, E, for a product p, E^p, is a function[2] of the average time $\bar{t}^p$, materials $\bar{m}^p$, and cost $\bar{c}^p$ required to make a product p. Scale manufacturing organises production to maximise

$$E^p_{\max} = \max_{\bar{c}^p, \bar{t}^p, \bar{m}^p} E^p(\bar{c}^p, \bar{t}^p, \bar{m}^p). \tag{1.1}$$

1.2 Definition (Manufacturing Flexibility)
Manufacturing flexibility F^p is a function of the average time $\bar{t}_r$ and cost $\bar{c}_r$ required to reconfigure production equipment in order to modify features of a product $p^i \rightarrow p^j : p^i, p^j \in P$, where p^i, p^j are variations of p and P is a set of products which can be made by a particular assembly system. Agile manufacturing organises production to maximise

$$F^p_{\max} = \max_{\bar{c}_r, \bar{t}_r} F^p(\bar{c}_r, \bar{t}_r). \tag{1.2}$$

The deployment of simple fast rigid dedicated assembly equipment tightly coupled to parts transfer equipment increases production effeciency E^p as a function of volume V^p increase with fixed production

[2]Normalisation and multi-dimensional scaling of $\bar{c}^p, \bar{t}^p, \bar{m}^p$ are required to define a consistent measure of E^p.

schedules. Production reconfiguration costs $F^p \propto V^p$ because of capital investment in dedicated equipment to generate economies of scale (Begg et al., 1994). V^p_* is an equilibrium volume for combined maximisation of E^p and F^p where E^p and F^p are combined linearly or nonlinearly. At V^p_* further reduction of production costs, i.e. further minimisation of E^p, is offset by escalating reconfiguration costs, i.e. reduction of F^p.

For a given production system, if equipment, processes, and products can be successfully concurrently engineered, efficiency and flexibility can be jointly maximised $EF^P_{\max}$. Figure 1.1 graphically represents a qualitative efficiency flexibility tradeoff for a hypothetical average production system containing nonlinear E^p and F^p. Figure 1.1 shows the intersection of an efficiency surface E with a flexibility surface F. E and F intersect at EF and $EF_{\max}$ indicates a unit cost and production time optimum corresponding to V^p_*. The zig-zag[3] line in Figure 1.1 indicates the impact on production system effeciency and flexibility as more advanced production technologies such as robot manipulators are deployed.

Design for automatic dis-assembly, recycling, and re-assembly is the highest design level strategy for increasing effeciency of material utilisation and flexibility of production to increase EF^p over the complete product range P. Robot manipulators are easier to reconfigure than special purpose equipment. Manipulators increase F^p. Manipulator control in flexible agile manufacturing systems is based on:

1. mathematical models whose parameters represent various aspects of a manipulator's operation sequences in its task space,
2. sensor measurements of task related features in workspaces.

An advanced manipulator workcell system typically comprises:

1. a manipulator geometrical structure,
2. a drive system, e.g. DC motors and transmission gearing,
3. an end-effector such as a gripper or welding tool,
4. sensors intrinsic to a manipulator such as optical encoders of joint positions,
5. sensors extrinsic to a manipulator:
   (a) contact vs non-contact (e.g. force vs visual),
   (b) active vs passive (e.g. laser range-finder vs steady visual),
6. extrinsic effectors such as part feeders,
7. controllers which link sensors to effectors,
8. man-machine interfaces such as GUI and voice command.

See (Craig, 1989; Paul, 1981; Fu et al., 1987). Assembly of products typically involves an observation phase when an object and its location and orientation are identified. This is followed by an operation phase when an object is picked up, relocated (while avoiding obstacles such as other manipulators) and mated to other parts or painted. If a manipulator is programmed to do high-level tasks, for instance to grasp a bolt moving on a conveyer belt and to insert it into a car chassis, this requires low

[3]The zig-zag line is a parametric function of historical time, indicated on one axis, as opposed to production time.

level software support for automatic planning of an obstacle-free path, automatic trajectory execution, possible on-line trajectory modification, automatic selection of a grasp location and real-time grasping of the bolt with compliant force-feedback control.

Research on robotic task models and matching of models to sensor measurements, see (Dorf, 1988), has had limited penetration into mainstream industrial robotics for reasons of economic cost, social resistance and technical complexity. Manipulators tend to be slower and more expensive than dedicated machinery. This reduces E^p. Manipulators also reduce E^p by wasting time changing tools or fixtures, adjusting to accommodate different operational phases, making measurements and decisions. Since manipulator acceleration and deceleration phases often consume most operation cycle time, these reduce E^p more than maximum end-effector velocity. Robot manipulators also require programming and re-programming time which further reduces effeciency. Usually it is difficult to derive adequate task models, for instance for visual object segmentation, without recourse to construction of special purpose workspaces. Even if adequate models and sensor processing algorithms for implementing individual task and operation units can be derived, the combinatoric complexity of possible assembly operations in moderately flexible manufacturing scenarios leads to complex hierarchical distributed software system engineering problems that are expensive to solve.

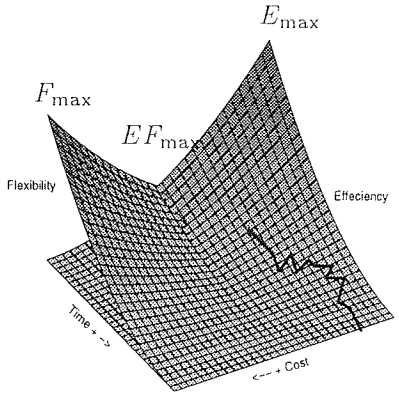

Figure 1.1: The Effeciency-Flexibility Tradeoff ($E^p \rightleftharpoons F^p$).

Formalisation of objective functions for optimisation of production organisational systems is also not a value-neutral activity (McCormick, 1981). Work cultural rigidities combined with technical myopia during design of large complex systems can sometimes reduce EF at great expense, for instance General Motors unsuccessful robotisation in the 1980s. The deployment of robots is also constrained by many general laws on the safety and reliability of computer software controlled machinery. See (Tapper, 1989). The commercial results of deployment of automation and robotic technologies are governed as much by social organisational factors as by mathematical, mechanical and computer engineering factors. See (Mansfield, 1989). Efficient social organisation of production and coherent interconnection of modular parts and flexible assembly processes is a prerequisite for the introduction of manipulators to translate to shorter lead-times from design to reconfiguration for flexible volume production.

Economic barriers and technical difficulties to automation are offset by increased competitiveness of agile flexible made-to-order manufacturing with advanced robotics and automation technologies. In the long term, failure to automate production results in industrial elimination. The thesis analyses a set of overlapping problems in distributed robotics and automation systems, software engineering, nonparametric regression and sensorimotor coordination that underlie development of more flexible agile manufacturing systems.

1.1 Manipulator Control Architectures and Software Engineering Methodologies

Since many technical engineering and production organisational factors are involved in successful development and deployment of advanced sensor and model-based robot systems, a central engineering problem is to derive a set of abstract robot control architectures. A robot control architecture is an ab-

stract organisation of software and hardware components into systems. Architectures are implemented with application domain frameworks that provide system developers with software components encapsulating computational services related to for instance process command, control, communication, and information transformations. Abstract architectures may be refined into more specific architectures for specific tasks and domains, e.g. tele-manipulation or assembly. Once an abstract architecture has been specified in an architecture specification language, design and implementation of a control system for a distributed sensor-actuator system such as a robot requires a methodology for organisation and coordination of software engineering. A systems engineering method is necessary to coordinate software engineers' concurrent integrated solution design, development, and implementation processes.

There are two basic design methodologies for definition of software architectures and corresponding organisation of engineering teams:

1. **functionalist** hierarchical parametric analysis and design,
2. **object oriented** analysis and design.

Functionalist systems analysis was the main engineering approach to design of robotic control architectures until the 1990s. See (Dorf, 1988; Fu et al., 1987; Brady, 1985). Specification, development and integration of software for robust functional control of manipulators that are easily re-programmable to perform a wide variety of sensor-based tasks has proven more difficult than early enthusiasts suspected. Defining control architectures for advanced tasks, such as outdoor fruit collection, involves engineering of complicated software that is difficult to debug and verify, especially sensor processing sub-modules.

Functional design of manipulator controllers involves top-down analysis of a task domain and synthesis of parametric models, data structures, and algorithms that operate on these models to perform tasks. After a task domain is analysed by a generic structured systems analysis and design method or by a specific structured functional software engineering methodology, software developers define and code appropriate explicit representations of manipulator mechanics and task domain properties. There are several specialist, e.g. NASA (Albus et al., 1987) and ESA (Elfving and Kirchhoff, 1991), software engineering methodologies for design of functionalist controllers of manipulators. The robot control design problem is a subset of the more difficult integrated automated manufacturing system design problem. Design for Automation (DFA) (Elfving and Kirchhoff, 1991) is a method for analysis of a task domain into elementary actions followed by synthesis of these into a control architecture which is used to structure program development around three conceptual reference models:

1. A Functional Reference Model (FRM) is a general model of control and information flows proposed for a variety of tasks.
2. An Application Reference Model (ARM) is an instantiation of a FRM for a specific application.
3. An Operations Reference Model (ORM) specifies the context in which an ARM functions. Context includes workspace and technician preconditions for ARM functioning.

A FRM has three top-down planning and control levels:

① A mission Level:

(a) involves spatial and temporal task reasoning in terms of manipulator primitive actions,

(b) requires symbolic logic planning and constraint satisfaction algorithms.

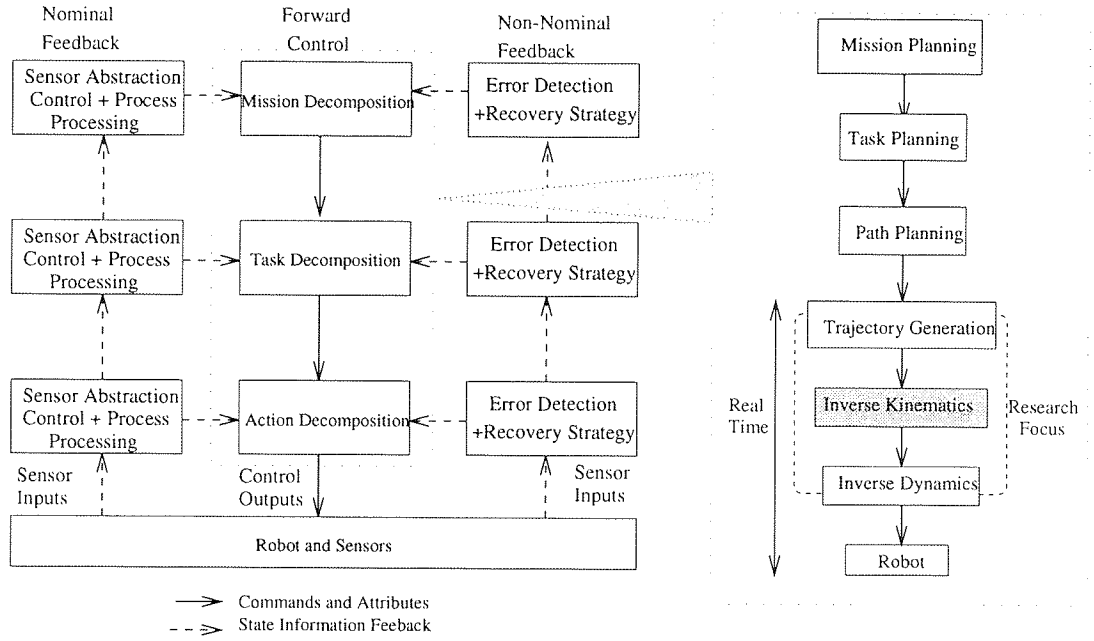

Figure 1.2: The Inverse Kinematic Level in a Functional Reference Model.

② A task level:

(a) involves spatial reasoning, path and grasp planning,

(b) requires sensor data fusion, computational geometry and database algorithms.

③ An action execution level:

❶ involves kinematic control of mechanical devices,

❷ involves dynamic control of mechanical devices,

❸ requires control (perhaps adaptive) and regression algorithms.

The three horizontal levels interact via three vertical control pathways:

1. Forward control specifies:

   (a) modules to activate in lower layers,

   (b) attributes for operation success and failure.

2. Nominal feedback relays back up attribute conditions.

3. Non-nominal feedback implements:

   (a) error detection,

   (b) error recovery.

See Figure 1.2. A FRM separates function and information flows from control actions. Once an ARM has been derived from a FRM and implemented, during system operation mission planning objectives are decomposed into task objectives which are translated into elementary action objectives which are executed. Once sequences of task objectives are defined, the lowest action execution level

involves intermediate kinematic and dynamic feedback control of a physical manipulator and primary sensor processing.

Kinematics is the systematic study of gemometric motions. Kinematic descriptions involve variables coding object position, velocity, acceleration and higher order derivatives without regard for forces which cause motion. The forward kinematics of a manipulator is a function

$$\mathcal{F} : \boldsymbol{\theta}_i \in \mathfrak{J}^n \rightarrow \mathbf{x}_i \in \mathfrak{C}^6 \equiv \mathsf{SE}(3) : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i) \tag{1.3}$$

from a coordinate system of manipulator joint angles to a coordinate system of the position and orientation of the manipulator end-effector, usually a gripper. $\mathcal{F}$ is usually a system of nonlinear transcendental equations. $\mathcal{F}$ may in certain cases be analytically solved to determine a set of inverse kinematic solutions for the inverse kinematic mapping $\mathcal{F}^{-1}$. Functional control architectures generally use analytical forward kinematic equations and analytical inverse kinematic solutions. Equation accuracy depends on measurement accuracy of equation parameters. High manipulator accuracy, up to $1mm$, to reach computed inverse kinematic positions, is important for functional model-based controllers. $\mathcal{F}^{-1}$ connects geometrical task databases to manipulators and their sensed task-spaces. Inverse kinematics also forms the basis of Free Space path planning. See (Latombe, 1991).

The forces that cause a given manipulator motion are determined by

$$\boldsymbol{\tau}_t = M(\boldsymbol{\theta}_t)\ddot{\boldsymbol{\theta}}_t + V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t) + F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t) + G(\boldsymbol{\theta}_t), \tag{1.4}$$

where $\tau_i^t \in \boldsymbol{\tau}_t$ are forces operating on each joint $\theta_i^t \in \boldsymbol{\theta}_t$ at time t. $M(\boldsymbol{\theta}_t)$ is a mass matrix. $V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t)$ are centrifugal and Coriolis terms. $F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t)$ is a friction term. $G(\boldsymbol{\theta}_t)$ is a gravity term. $V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t), F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t)$ are difficult to model. For small $\dot{\boldsymbol{\theta}}_t$, $V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t), F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t)$ are usually neglected to construct an inverse dynamic controller $\mathcal{F}_\tau^{-1}$

$$\mathcal{F}_\tau^{-1}(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t, \ddot{\boldsymbol{\theta}}_t) \approx M(\boldsymbol{\theta}_t)\ddot{\boldsymbol{\theta}}_t + G(\boldsymbol{\theta}_t). \tag{1.5}$$

To compensate for measurement quantisation errors and computation delays, analytical inverse control models are supplemented with adaptive controllers. A set of linear adaptive controllers, for instance Proportional Integral Derivative (PID) for each θ_i^t, work well at low $\dot{\boldsymbol{\theta}}_t$ where link interactions $V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t), F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t)$ are negligible. Accuracy requirements of control execution are high and large amounts of information and control commands have to be processed in real-time.

During task execution, state information from sensors flows back up the ARM structure. Nominal feedback relays sensor state information and intermediate processing information up a hierarchical control structure. This enables checking for completion of actions at lower levels or modification of internal model representations. If non-nominal feedback detects forward control errors, pre-programmed recovery strategies are activated. Functionalist manipulator control generally relies on precise models of a manipulator and a task space, sometimes in a form amenable to matching with sensor data. In structured workspaces such as factories, much Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) and Computer Integrated Manufactruing (CIM) data is available about a manipulator and its assembly tasks. Multi-format CAD data generated by concurrent and pipelined engineering design teams can be integrated by conversion into the EXPRESS language of the Standard for the Exchange of Product model data (STEP) (Owen, 1993). Definition of appropriate CAD-based task representation for construction of internal models based on a-priori information enables complex conditional branching manipulator assembly sequences to be programmed if signal data can be processed to trigger branch execution conditions. This reduces labour factor costs, increases manufacturing flexibility, and shortens the time pipeline from product design to prototyping to production of customised one-off variants made to order.

Environmental feedback information on which mission level control is based is low abstraction, high granularity and subject to distortion by sensor quantisation and other types of noise. It is usually difficult (Fu et al., 1987; Vernon, 1991; Horn, 1986) to process signals, for instance images, to extract consistent invariants (for instance geometrical, colour, motion) corresponding to task-objects. An object recognition invariant is a property of noisy signals that distinguishes one object from a set of objects. A set of object recognition invariants forms a basis for classification of signal subsets as probabilities of the presence of objects generating signals. STEP representations of part assembly geometry and constraints provides a foundation for large scale flexible Bayesian pattern recognition support technologies for rapid robotic assembly reconfiguration of CAM/CIM systems as well as the usual workcell design and robot programming. However, it is a difficult task to extract and merge variable granularity CAD data for Bayesian pattern recognition. It is difficult to merge 2D and 3D wireframe CADs, exact solid CAD, surface CAD, shape deformation CAD, geodesic CAD, finite element CAD, and assembly modeling CAD representations. Dispersed CAD data centralisation may also be a difficult data warehouse problem spanning design and production processes.

Object recognition invariants are required for correct operation of geometrical and propositional calculus workspace models operated on by spatial and task planning software (See for instance (Fu et al., 1987)). Development of low-level signal processing software for computing invariants that span a functional task space is thus essential for calibration of functional control architectures to general and specific task-spaces. Machine vision (Horn, 1986; Vernon, 1991) is limited in its capacity to abstract categorical invariances (see (Lee, 1991b; Horn, 1986) for an analysis of problems with image processing) for mission planning symbolic logic planning software to operate on. Symbolic logic compilers are a useful descriptive level for programming arithmetic algorithms to achieve certain tasks, for instance constraint satisfaction in industrial assembly and process control. Language compilers like Prolog enable hard problems which are difficult to explicitly formulate to be solved. Expert Systems for error diagnosis and recovery planning increase the functionality of DFA robotic work cells. Fuzzy logic compilers may be useful to enable programmers to formalise qualitative strategic levels of functionalist architectures into discrete mathematics. Linking of codes produced by qualitative reasoning compilers with explicitly quantitative numerical and statistical codes, e.g. (Press et al., 1992), increases overall DFA system robustness.

The cost and unreliability of DFA style software combined with operator training costs are the main reasons why simple teach and show robots predominate in factories. The limitations of complicated inflexible ARM software that is difficult to develop, program, and reprogram for new industrial assembly scenarios were a counter incentive to capital investment in advanced industrial robotisation during the 1980s. In the 1990s, object oriented analysis and design system engineering methodologies, e.g. (Booch, 1994; Rumbaugh, 1991; Break and Haugen, 1993; SDL, 1993; Cook and Daniels, 1994) are penetrating the industrial robotics and automation mainstream. See (ITU/ISO/IEC, 1996; Bender, 1993; Siegel, 1996; Usländer, 1995b; Usländer, 1995a; Owen, 1993). Object analysis and design methodologies are an extension of structured functional analysis and design methodologies for engineering of large systems. The principal difference between object and functional design methodologies is that object methodologies group data and data transformation operations into hierarchical modules whereas functional methodologies have a global data model which is operated on by hierarchically structured functions of an ARM. An optimal of an object system design maximises interlocking component design representational effeciency, dependability, durability, reusablity, extendability and maintainability.

While object analysis and design is not a panacea for systems engineering since object systems

are difficult to analyse, system integrators such as GEC, Siemens, Bosch, Mitsubishi, Sumitomo, Alcatel, and ABB now build their robotics and automation architectures with objects that can inter-communicate across standard CAD/CIM representation conversion standards such as STEP (Owen, 1993) over standard CIM protocols such as Profibus (Bender, 1993). Profibus has three communication protocols operating at three hardware spatio-temporal resolutions:

1. the fieldbus level protocol has a maximum $10ms$ bus cycle time for time critical processes
2. the workcell level protocol has a maximum $100ms$ bus cycle time for process control.
3. the systems level protocol has a maximum $1000ms$ bus cycle time for TCP/IP inter-connection.

CIM communication and control protocols such as Profibus support hierarchical distributed real-time control of multi-robot, multi-part-feeder, multi-sensor configurations. Robot manipulators are elements of distributed manufacturing systems. Open distributed agile manufacturing systems require coordination of hundreds of thousands of distributed devices and computational objects. There are many problems to implement high performance distributed robotics and automation systems with soft real-time constraints of high system availability and reliability or hard fault-tolerant constraints. See (Mullender, 1989). Developers require tools and library components built on international standards to easily implement and reconfigure secure reliable mixed synchronous and asynchronous transaction based multi-tier architectures with system management and quality of service interfaces.

Current integrated CAD/CAM/CIM object based industrial robotics and automation systems provide the substrate for advancing sensor and model based agile single lot made-to-order design and production. A distributed object environment for point and click image-based tele-command of manipulators and stereo-cameras is specified and implemented in Chapter 6 that supports rapid prototyping of numerical experiments on control of distributed cameras and robot manipulators. The implementation experience leads to a specification in Appendix B of key components of the mapping of the Common Request Broker Architecture (Siegel, 1996) components and services to the Open Distributed Processing Reference Model technology level (ITU/ISO/IEC, 1996). This leads to a specification in Chapter 6 of design and performance requirements for scalable virtual micro-kernels that broker inter-Java-virtual-machine remote method invocations.

The complexity of a given ARM, including object ARM, for sensor-based manipulator control is constrained at the action level by the limits of signal processing algorithms for extraction of low-level invariant features from images. These limits constrain what it is possible to engineer at the mission level based on object classification. Successful application of the DFA method require:

1. a structured manipulator work-cell,
2. an accurate manipulator with exact kinematic description,
3. exact parametric models of task objects and obstacles,
4. large free spaces,
5. large tolerances for manipulator operations or special tools and peripheral devices,
6. pre-programming and simulation.

The results of functional and object system analysis depend on the quality and extent of system designers' analyses and syntheses of representations at the architectural, application, and operation

levels (e.g. FRM, ARM, and ORM). DFA in particular is engineering labour intensive and expensive. In practice for economic and software engineering complexity reasons, DFA model and task representations are often tightly coupled to details of parts conveyance and kinematic and dynamic characteristics of particular models of manipulators. The primary disadvantage of DFA is that if representations of a manipulator kinematics or its workspace are incorrect, then actions based on them are incorrect. This can lead to operational inflexibility. Typically in industrial assembly, at task set initialisation, a manipulator is calibrated to perform a series of tasks in a workspace. DFA based systems fail if there is

① substantial uncertainty in measuring part positions and features due to bounds on sensor resolution,

② limited and variable manipulator kinematic and dynamic precision in a workspace,

③ a discrepancy between internal models and an external workspace.

These conditions result in collisions and failed grasping. Over time, if discrepancies between internal models (for instance kinematic equation coefficients) and an external workspace increase due to wear and tear on a manipulator, or if a workspace changes between initial modeling and plan execution, re-calibration of manipulator kinematics or task-space by extraneous repositioning of objects to suit representations is easier than reprogramming.

1.2 Calibration of Control Software Coordinate Systems to Manipulators

Calibration and re-calibration of manipulator kinematic equation parameters and parameters governing a mapping of sensor measurements to the coordinate system of kinematic equations is a basic task underlying functionalist programming of manipulators (Craig, 1989; Fu et al., 1987). While some research, for instance on compliant grasping (Fu et al., 1987), attempts to make parametric model based control of manipulators more robust to precision uncertainty, robust control methods require a minimum level of calibration for gross task positioning before engaging sensor servoing modes for manipulation in the presence of sensor uncertainty.

Manipulators whose operation is based on complex internal parametric models for assembly must have these models calibrated to objects in the workspace by sensor measurements and must also have their inverse kinematics calibrated to the workspace and to the task objects. Before object models are calibrated to sensor data, a manipulator must be calibrated to its workspace. Depending on manufacturing tolerances, calibration of the sensors to the workspace may be required daily at manipulator initialisation. Without correct kinematic model calibration, higher strategic level geometric and task planning models, assembly scheduling, and error recovery sub-systems will fail since models are generally based on exact representations. During manipulator operation, if de-calibration is sufficiently radical, on-line calibration may not be possible. In this case off-line re-calibration time should be minimal in order not to reduce efficiency of assembly operations.

Manipulator kinematic structures are usually represented with parameters in a Denavit Hartenberg (DH) coordinate system. After sequential calibration of stereo-camera and DH parameters, in order to accelerate exact Newton second-order re-calibration of DH parameters, an optimal numerical evaluation of DH matrix first order and second order error derivatives with respect to a re-calibration error function is derived, implemented and tested in Chapter 3.

Simultaneous calibration of a composition of stereo-camera and manipulator kinematic models is under-determined by equi-output parameter groups in the composition of stereo-camera and Denavit Hartenberg (DH) models. In Chapter 4, an error measure for simultaneous calibration of a composition of models is derived and parameter subsets with no equi-output calibration solution groups are determined by numerical experiments to simultaneously calibrate the composition of homogeneous or pan-tilt stereo-camera with DH models.

Adaptive sensor-based functional control enables more robust manipulation of known objects (e.g. screws) in slightly dislocated or un-prespecified positions and orientations, manipulation of known but slightly deformed objects, and high level point and click tele-manipulation of unknown objects. Remote or local technicians can tele-interact with algorithms at mission and task levels. For remote manipulator operations, for instance low earth orbital or inter-planetary, transmission delays require semi-autonomous operation during compliant motion and kinematic trajectory inversion. Functional and object control architectures with representational adaptivity increase robustness of overall system operation.

The limitation of teach and show programming is that since no sensors are involved, no task space measurements are made. Hence there can be no interpolation or extrapolation to different or new tasks. The teach and show approach is still attractive in principle if it may be extended by incorporation of sensors, for instance cameras, to more complicated tasks, by use of statistical regression and decision theory, flexible nonparametric task representations, and deductive and inductive inference. The main statistical models of learning that have been developed over the last one hundred years are (Thorndike, 1949)'s connectionism, (Pavlov, 1927)'s classical conditioning, (Guthrie, 1952)'s contiguous conditioning, (Tolman, 1932)'s sign learning, (Hull, 1943)'s systematic behaviour theory, (Skinner, 1953)'s operant conditioning, (Piaget, 1972)'s developmental groups (see (Hilgard and Bower, 1975; Chaplin and Krawiec, 1979) for integrated analyses of psychological learning theories) and neural network models (Rosenblatt, 1962; Rumelhart and McClelland, 1986; Rohwer, 1994). Neural network models can be used for nonparametric regression and pattern classification. If the application of nonparametric regression in an assembly domain can reduce the programming required for a robot to perform a new set of tasks, then it has been hypothesised by a few researchers that nonparametric regression components have applications in manipulator control systems.

1.3 Calibration of Nonparametric Models of Manipulators

Nonparametric models can approximate several classes of robot manipulator control functions. See for instance (Warwick et al., 1992; White and Sofge, 1992; Sanner and Slotine, 1995; Miller III et al., 1990; van der Smagt, 1994; Mel, 1990; Miller, 1989; Miller et al., 1990; Cooperstock and Milios, 1993; Walter and Schulten, 1993). Nonparametric classification (Fukunaga, 1990; Duda and Hart, 1973; Devijer and Kittler, 1982; Bishop, 1995; Ripley, 1996) and nonlinear multi-variate regression (Seber and Wild, 1989; Mardia et al., 1979) theory and methods provide a framework for determination of optimal nonparametric model order and parameter values for regression on manipulator functions. Regression functions are defined on vector spaces of representable functions ϕ and random functions ϖ

$$\mathbf{x}_i = \phi(\boldsymbol{\theta}_i) + \varpi_i. \tag{1.6}$$

Regression methods fit models to data assumed to be generated by regression functions by minimisation of various error functions in order to approximate ϕ. Multi-variate nonlinear regression on finite

data involves a collection of methods to fit finite dimensional representable functions to finite noisy data sets assumed to be generated by random functions after two assumptions have been made:

1. model identification - a sub class of representable functions ϕ must be selected which satisfy certain error criteria for a fit to a data set.

2. noise identification - the properties and form of the noise component of the data set must be determined. For instance ϖ is a random variable with an associated distribution, for instance Gaussian.

There are two basic types of nonlinear regression models - parametric and nonparametric.

1.3 Definition (A Parametric Model)
A parametric model is a parameterised class of functions which have been identified a priori during a system identification phase, for instance a set of kinematic equations.

1.4 Definition (A Nonparametric Model)
A nonparametric model is a general function whose parameters do not correspond to identifiable elements of a modelling domain. A nonparametric model is usually a class of kernel expansions used in regression. Feed-forward networks and radial basis functions are nonparametric models. See (Bishop, 1995).

Parametric and nonparametric models may be combined in order to define semiparametric regression.

1.5 Definition (A Semiparametric Model)
A semiparametric model is a combination of parametric and nonparametric models. This may take two forms:

1. A suitable partition of a regression domain between models.

2. A combination of models over the same regression pre-image.

Combination of parametric models and nonparametric models into a semiparametric model may gain the advantages of efficiency of analytical functional modelling and flexibility of nonparametric approximation if an analytical function is incomplete or partially incorrect.

Nonparametric regression, for instance general linear regression (Sanner and Slotine, 1995), can model $V(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t), F(\boldsymbol{\theta}_t, \dot{\boldsymbol{\theta}}_t), M(\boldsymbol{\theta}_t)$ nonlinearity terms of (1.4) in high acceleration heavy payload relative to arm mass control regimes. This work focuses on the kinematic level of analysis of the manipulator control hierarchy. See Figure 1.2. Solutions for manipulator kinematics based on behavioural regression (see (Skinner, 1953)) on nonparametric models have been proposed by for instance (Ritter et al., 1992; Walter and Schulten, 1993; Kuperstein and Rubinstein, 1989; Edelman, 1989; Miller, 1989; Hashimoto et al., 1992). This thesis contains a comparison of analytic and nonparametric approximation of solutions of manipulator kinematic functions and an analysis of camera and manipulator kinematic model calibration issues. A prototype distributed object manipulator control environment is validated by a hierarchical k-fold cross validated calibration to Cartesian space of a radial basis function regression correction of an affine stereo model in §5.5.

Kinematic inversion efficiency is important to satisfy soft or hard real-time constraints on kinematic trajectory inversion. If a nonparametric regression system for complete inverse kinematics is implemented in special purpose hardware, it may be computationally more efficient than a parallelised analytical solution such as (Lee, 1991a; Zhang and Paul, 1991).

The kinematic mapping of a rigid open-link manipulator is a homomorphism between Lie groups. The homomorphism has solution groups that act on a solution element. A geometric analysis of the

inverse kinematic solution group of three and seven degree-of-freedom (*dof*) dextrous manipulators results in the determination of a canonical representation of solution group operators that act on a solution element. This is a collection of trivial fibre bundles with solution fires homotopic to the Torus. If fibre solutions are parameterised by a scalar, a direct inverse function (and an inverse augmented Jacobian) that maps Cartesian base space coordinates (differentials) to the solution element fibre coordinates (differentials) is defined. Since it is difficult to solve analytically for seven *dof* arms, the direct inverse function can be approximated by a tessellation of Cartesian base space by first order Taylor expansions where the first order term is the inverse augmented Jacobian, i.e. a local linear approximation. A combination of canonical solution group operators and a local linear approximation of the solution element of inverse kinematics of dextrous manipulators is a cyclic and parallelisable solution. Solution query evaluation time is bounded above by retrieval speed from RAM cache of Taylor expansions that bound the hyper-cube base subspace of a solution query. The solution group operators and local linear approximation of solution element is proposed as a theory of representation of kinematic transformations in primate nervous systems.

1.4 Summary of the Research Objectives and the Thesis Contents

The research foreground is generated by an intersection of several background knowledge domains: functionalist and object oriented distributed software system engineering, solution groups of homomorphisms, manipulator inverse kinematics, model calibration, nonlinear nonparametric regression and sensorimotor coordination theory. Some background domain theory is abstracted in the foreground in order to construct the foreground intersection space where the foreground research objectives ordered by priority are to:

1. propose a theory of representation of arm inverse kinematic transformations in primate nervous systems. See Chapter 9 and §9.4.

2. analyse the canonical solution groups of a homomorphism from the n-Torus to the Special Euclidean Group. See §7.7 and Chapter 8.

3. derive a canonical representation of solution groups and a local linear approximation for the solution elements of Planar 3 *dof* and 7 *dof* dextrous manipulators. See §8.2 and §8.3.

4. derive and analyse error measures for simultaneous calibration of DH and stereo-camera parameters. See §4.1.

5. analyse and determine equi-output parameter groups of the composition of stereo-camera and DH models. See §4.2 and §4.3.

6. derive an optimal re-calibration of DH Parameters with numerical products of DH matrix derivatives. See §3.3.

7. compare and contrast representation and calibration of analytical and nonparametric representations of stereo-cameras and spatial and dextrous rigid manipulator open-chain kinematic functions. See Chapters 2, 3, and 4 and §5.3.

8. implement an experiment for hierarchical k-fold cross validation of a radial basis function regression correction of an affine stereo model of real cameras viewing a physical manipulator. See §5.5.

9. specify and implement a distributed object environment for prototyping of manipulator control algorithms that incorporates a graphical tele-command interface to physical robots and stereo-cameras that supports point and click on stereo-image based manipulator kinematic command. See §6.1.

10. specify a mapping from the Common Request Broker Architecture to the Open Distributed Processing Reference Model technology level. See Appendix B.

11. specify non-functional requirements for a scalable virtual distributed kernel event transport system. See §6.5.

Chapter 2

Inverse Kinematic Solutions of Rigid Open-Chain Manipulators

This Chapter compares and contrasts methods for representing inverse kinematics of redundant and non-redundant manipulators. A rigid manipulator is comprised of a chain of links connected by joints. A link is a rigid body defining the spatial relationship between two neighbouring joint axes. Joints are capable of rotational **R** or translational **P** movement. Manipulator kinematics describes geometric and time based properties of link coordinate system movements with respect to one another and with respect to a workspace $\mathfrak{W}$ inertial coordinate system. Manipulator links can be arranged as open or closed chains. Closed chain structures such as a Stewart mechanism (c.f. (Craig, 1989)) have joints that connect more than two links. A closed chain increases manipulator stiffness, hence speed and efficiency for dedicated tasks at the cost of reduced $\mathfrak{W}$ range and task flexibility. Most manipulators are open chains of joints, each with one degree of freedom (*dof*) to move.

2.1 The Homogeneous Representation of Kinematic Transformations

For an open-chain manipulator with n joints numbered 1 to n, there are $n+1$ links, numbered from 0 to n. Link 0, the base of the manipulator, is fixed. Link n carries an end effector. Joint i connects link i to link $i-1$. A link may be specified by two parameters, length a and twist α, which define the relative location of the two joint axes that a link connects in space. Link parameters for the first and last links are set to 0. Joints may be described by two parameters, offset d, and angle θ. Parameter θ is variable for **R**, while parameter d is variable for **P**.

link	joint	twist	length	offset
i	θ_i^o	α_i^o	a_i(mm)	d_i (mm)
1	θ_1	-90^0	0	0
2	θ_2	0	431.8	149.09
3	θ_3	90	-20.32	0
4	θ_4	-90^0	0	433.07
5	θ_5	90^0	0	0
6	θ_6	0	0	56.25

Table 2.1: The Denavit Hartenberg Table of a PUMA 560.

The geometry of a manipulator is defined by the way the links of a manipulator are connected together. To describe the position and orientation of a link with respect to other links, a rectilinear coordinate system or frame is affixed to each link, frame i attached to link i. Denavit Hartenberg (DH) is a convention for attaching coordinate reference frames to links. (Craig, 1989) and (Fu et al.,

1987) are two common DH conventions. Neither applies to closed chains. The (Fu et al., 1987) DH[1] convention is described in this section since it simplifies gradient and Hessian evaluation on pg. 61.

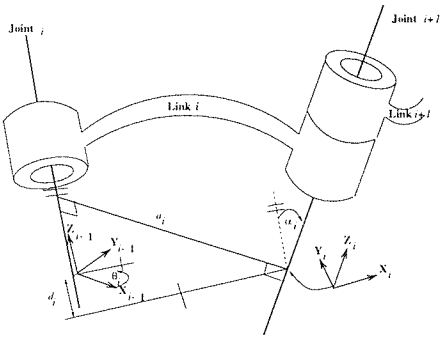

Figure 2.1: The (Fu et al., 1987) Denavit Hartenberg Convention.

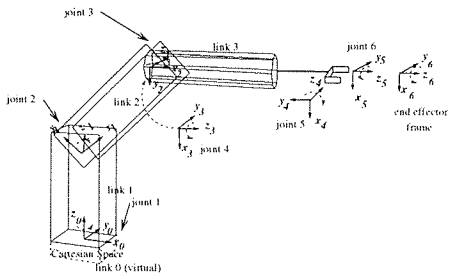

Figure 2.2: The Assignment of Coordinate Frames to an Anthropomorphic Manipulator.

In the (Fu et al., 1987) DH convention, the orthogonal basis for a given coordinate frame $\{A\}_i$ is denoted $\{X_i, Y_i, Z_i\}$. Z_{i-1} is aligned with the axis of joint i. The X_{i-1} axis is directed along the normal from Z_{i-1} to Z_i and for intersecting axes is parallel to $Z_{i-1} \times Z_i$. Frame assignment is graphically illustrated in Figure 2.1 where

a_i is the link **length** between the Z_{i-1} and Z_i axes along the X_{i-1} axis.

α_i is the **twist** angle between the Z_{i-1} and Z_i axes about the X_i axis.

d_i is the **offset** distance along Z_{i-1} between the X_{i-1} and X_i axes.

θ_i is the joint **angle** between the X_{i-1} and X_i axes looking down the Z_{i-1} axis.

Figure 2.2 shows a sequence of coordinate frame assignments to an anthropomorphic manipulator. The Cartesian frame, located at the base of a manipulator is denoted $\{X_0, Y_0, Z_0\}$, with Z_0 always by convention collinear with Z_1. d_0, d_1, a_0, and α_0 are always 0.

Once a frame assignment convention is chosen and link ($\mathbf{a}$, $\boldsymbol{\alpha}$) and joint ($\mathbf{d}$, $\boldsymbol{\theta}$) parameters are measured, a manipulator is kinematically specified. It is usual to organise a specification into a DH Table. Table 2.1 represents the geometry of a PUMA 560.

The position and orientation of frame $\{A\}_i$ with respect to $\{A\}_{i-1}$ can be computed by a series of four transformations of frame $\{A\}_{i-1}$:

1. Rotation about the Z_{i-1} axis by an angle of θ_i to align the X_{i-1} axis with the X_i axis.

2. Translation along the Z_{i-1} axis a distance of d_i brings the X_{i-1} and X_i axes into coincidence.

3. Translation along the X_i axis a distance of a_i brings the two origins as well as the X axes into coincidence.

4. Rotation about the X_i axis by an angle of α_i brings the two coordinate systems into coincidence.

[1]DH used in mathematical formulas, e.g. (2.12), refers to the parameter matrix resulting from a DH convention.

These four transformations may be represented by rotation or translation matrices which when multiplied together form a homogeneous transform from $\{A\}_{i-1}$ to $\{A\}_i$

$$
{}^{i-1}_{i}\mathbf{T} \equiv \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.1}
$$

$$
\equiv \begin{bmatrix} & {}^{i-1}_{i}\mathbf{R} & & {}^{i-1}\mathbf{O}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.2}
$$

${}^{i-1}_{i}\mathbf{R}$, the upper left 3×3 sub-matrix of (2.1), defines the rotation of $\{A\}_i$ relative to $\{A\}_{i-1}$. Thus, ${}^{i-1}_{i}\mathbf{R}$ is a Special Orthogonal matrix (See §7.4). ${}^{i-1}\mathbf{O}_i$ the upper right 3×1 sub-matrix translates the origin of frame $\{A\}_{i-1}$ to frame $\{A\}_i$. Homogeneous transforms are defined in a homogeneous coordinate system $\{x, y, z, s\}$ which is an embedding of a $\mathbb{R}^3$ manifold with s a scaling parameter. The internal structure of a homogeneous transform is useful to separate rotational and translational components of kinematic transformations. The bottom row of a homogeneous transform is used for scaling and perspective transforms. In (2.3), c is a scaling factor, f is focal length parameter which governs a perspective projection $\mathbf{P}$

$$
\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & \frac{1}{-f} & c \end{bmatrix}, \tag{2.3}
$$

where a, b are radial distortion factors. See (4.12) and Figure 3.2. Combined rotation and translation of a frame results in a screw motion. Homogeneous transforms are screw operators. See Definition 7.32 and (Samuel et al., 1991).

Computationally efficient representations of rotations of frame $\{A\}_{i+1}$ w.r.t $\{A\}_i$ are sometimes used to derive more efficient representations of inverse kinematics for inverse kinematic trajectory inversion (see §2.4.1):

1. Fixed angle conventions: successive rotations around $\{A\}_i$ frame axes $\{X_i, Y_i, Z_i\}$ by γ, ω, λ.
2. Euler angles: successive rotations around $\{A\}_{i+1}$ frame axes $\{X_{i+1}, Y_{i+1}, Z_{i+1}\}$ by γ, ω, λ.
3. Equivalent angle-axis: Specification of a non-frame axis $\hat{X}$ and a rotation angle α.
4. Quaternions. See Definition 7.34. Quaternion algebra is defined on scalar a and vector $\mathbf{b}$ elements such that

$$
\mu = (a, \mathbf{b}) = x_1 + x_2\mathbf{i} + x_3\mathbf{j} + x_4\mathbf{k} \quad \begin{cases} a \in \mathbb{R} & (x_1), \\ \mathbf{b} \in \mathbb{R}^3 & (x_2, x_3, x_4). \end{cases} \tag{2.4}
$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are right handed orthormal components of vectors in $\mathbb{R}^3$.

2.2 A Classification of Manipulator Geometries

A manipulator chain that can grasp non-oriented objects in $\mathfrak{W}$ is a regional manipulator. This requires a minimum of 3 *dof*. A chain that can grasp an object in $\mathfrak{W}$ at arbitrary position and orientation is

Manipulator	Simulated	Real
Planar 2-**R**	✓	
Planar 3-**R**	✓	
PUMA 560	✓	
SCARA	✓	✓
RR-K1207	✓	

Table 2.2: The Manipulators Analysed in the Thesis.

link chains	joints types	*dof* for tasks
open	rotational(**R**)	regional $\Longrightarrow$ 3
closed	prismatic(**P**)	spatial $\Longrightarrow$ 6

Table 2.3: Types of Manipulator Chains.

a spatial manipulator. A spatial manipulator requires a minimum of 6 *dof*: 3 *dof* for positioning and 3 *dof* for orienting the manipulator. See Table 2.2.

A kinematically redundant manipulator has more *dof* than are required for a given task. If redundant (> 6) *dof* are added to spatial manipulator, it is dextrous. Restricted $\mathfrak{W}$ and specialised tasks require manipulators with less *dof*, viz. painting manipulators need a maximum of 5 *dof*, and planar manipulators may be adequate for some types of sheet metal cutting.

Many models of open chain rigid manipulators that satisfy various engineering design criteria have been fabricated for industrial applications. Industrial open-chain spatial manipulator models are classified according to the design of the positioning structure (first 3 *dof* from base of a manipulator) (Craig, 1989). Manipulator geometries can be:

1. (**2R-R**) **Anthropomorphic** if a manipulator has a **2R** shoulder and an **R** elbow. Anthropomorphic designs, for instance the PUMA 560 (see Table 2.1) can reach into confined spaces.

2. (**2R-P**) **Spherical** if a manipulator is similar to anthropomorphic manipulator except the elbow is a **P**.

3. (**P-P-P**) **Cartesian** if a manipulator has 3 mutually orthogonal **P** joints corresponding to (x, y, z) coordinates in $\mathfrak{W}$. The relation of **P** values to position in $\mathfrak{W}$ is an identity mapping.

4. (**P-R-P**) **Cylindrical** if a manipulator uses a **P** to translate the arm vertically, has an **R** with a vertical axis and a **P** that is orthogonal to the **R** axis.

5. (**P-3R**) **SCARA** if a manipulator has one base **P** and three parallel **R**. SCARA design was motivated to simplify kinematic and dynamic analysis of interaction of payloads, friction and vibration but there are operational holes in a workspace $\mathfrak{W}$. **R** joints do not support any weight and link 0 houses all heavy actuators so a SCARA can be large and fast.

This work considers the kinematics of the manipulators in Table 2.2.

2.3 Forward Kinematic Functions

For an open-chain manipulator $\mathfrak{m}$ with n joints, its n homogeneous frame transforms may be multiplied together to produce a single transformation ${}^0_n\mathbf{T}$ that relates frame $\{A\}_n$ to frame $\{A\}_0$

$$ {}^0_n\mathbf{T} = {}^0_1\mathbf{T}\,{}^1_2\mathbf{T}\ldots{}^{n-1}_n\mathbf{T}. \tag{2.5} $$

A homogeneous representation of points enables one to switch back and forth between vector and matrix notations for points. The position and orientation of a spatial manipulator end-effector is a point in a six dimensional space. This point can be represented by a homogeneous matrix following

a DH convention for instance. If $\boldsymbol{\theta}_i$ represents a point of arbitrary dimension in a pre-image of a mapping $\mathcal{F}$ and $\mathbf{x}_i$ represent a point of arbitrary dimension in the image of $\mathcal{F}$, from §2.1 for $\mathfrak{m}$ with only[2] $\mathbf{R}$, ${}^0_n\mathbf{T}$ is a nonlinear function of all n joint variables $\boldsymbol{\theta}_i$ and

$$ {}^0_n\mathbf{T} = \mathcal{F}(\boldsymbol{\theta}_i). \tag{2.6} $$

2.1 Definition (A Forward Kinematic Mapping)
${}^0_n\mathbf{T}$ is a representation of the forward kinematic[3] mapping $\mathcal{F}$ of a manipulator

$$ \mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{C}^m : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i). \tag{2.7} $$

The pre-image[4] of the mapping is the n-torus T^n. The C^m image of the mapping is known as Cartesian Space. See §7.4. It is convenient but not necessary to represent $\mathcal{F}$ by (2.6).

2.2 Definition (Cartesian Space)
In the case of a spatial or dextrous manipulator, C^m is the product space of rigid body rotations by translations in a subset of 3 dimensional physical space. The points of the image are configurations of a manipulator. A configuration is a rigid body transformation, or positive isometry, i.e. a distance and orientation preserving transformation. A configuration space C^m is the set of all possible configurations. See Definition 7.32.

2.3 Definition (Joint Space)
The pre-image of the kinematic mapping $\mathcal{F}$ is the n-torus T^n. This is known as joint space.

In order to compute the forward kinematics of a given manipulator it is necessary to represent the n-torus and manipulator configuration space with particular coordinate systems.

2.4 Definition (Cartesian Coordinates)
A representation of a configuration of a manipulator is a set of coordinates $\{c_1, \ldots, c_n\}$ specifying position and orientation of a manipulator with respect to some reference frame, usually the Cartesian frame $\{X_0, Y_0, Z_0\}$. In this case, $\mathfrak{C}$ the Cartesian coordinate system is defined.

2.5 Definition (Joint Coordinates)
A representation of the joint values of the manipulator is specified with respect to a DH convention. A joint coordinate system defined by a DH convention is denoted $\mathfrak{J}$.

(2.7) is defined on $\mathsf{T}^n \rightarrow \mathsf{C}^m$ in the abstract but for inverse kinematics of a real manipulator, $\mathcal{F}$ is defined on $\mathfrak{J}$ and $\mathfrak{C}$ representations of T^n and C^m respectively when this is not explicitly indicated.

Sometimes in robotics, a representation of a Cartesian space, i.e. a Cartesian coordinate system $\mathfrak{C}$, is itself called Cartesian space. In this work also, occasionally $\mathfrak{C}$ is called Cartesian space. When Cartesian space has the structure of the Special Euclidean Group, occasionally also $\mathfrak{C}$ and SE(3) are used interchangeably in order to emphasise that $\mathfrak{C}$ has translational and rotational structure.

From (2.7), if $n > m$, $\mathfrak{m}$ is redundant by $n - m$ degrees. Although redundancy is task dependent, manipulators with more than 6 *dof* are usually called redundant or dextrous since end effector positioning and orientation can be handled by a 6 *dof* spatial manipulator. For spatial ($n = 6$) and dextrous ($n > 6$) manipulators, a forward mapping is defined

$$ \mathcal{F} : \boldsymbol{\theta}_i \in \mathsf{T}^n \rightarrow \mathbf{x}_i \in \mathsf{C}^m \equiv \mathsf{SE}(3) : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i), \tag{2.8} $$

[2] For $\mathfrak{m}$ with $\mathbf{P}$ joints, ${}^0_n\mathbf{T}$ is a nonlinear function of $\mathbf{d}$.

[3] Contrary to open-chains, computation of forward kinematics $\mathcal{F}$ is difficult for closed-chains while inverse transformation $\mathcal{F}^{-1}$ is trivial (Samson et al., 1991).

[4] In this work, n and m generally refer to the dimensionality of the pre-image and image of a kinematic mapping (2.7).

between manifolds of dimensionality n and 6 respectively.

$$\begin{array}{rcccccc} x_1 &=& f_1(\theta_1, & \theta_2, & \theta_3, & \ldots, & \theta_n), \\ x_2 &=& f_2(\theta_1, & \theta_2, & \theta_3, & \ldots, & \theta_n), \\ \vdots &=& \vdots & \vdots & \vdots & \ddots & \vdots \\ x_6 &=& f_6(\theta_1, & \theta_2, & \theta_3, & \ldots, & \theta_n). \end{array} \tag{2.9}$$

A forward kinematic mapping $\mathcal{F}$, a smooth ($\mathcal{C}^\infty$) map from T^n to SE(3), can be differentiated.

2.6 Definition (A Jacobian)
A Jacobian of a kinematic mapping $\mathcal{F}$ is a second order tensor, a $m \times n$ matrix of derivatives $\mathbf{J}$, that maps velocities $\dot{\boldsymbol{\theta}}_i$ in the tangent bundle (See §7.2) of T^n to velocities $\dot{\mathbf{x}}_i$ in the tangent bundle of C^m at $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$. For $\mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{C}^m$, $\mathbf{J}$ is defined as $\dot{\mathbf{x}}_i = \mathbf{J}\dot{\boldsymbol{\theta}}_i$ defined component wise and omitting i subscript as

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \cdots & \frac{\partial f_2}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \theta_1} & \frac{\partial f_m}{\partial \theta_2} & \cdots & \frac{\partial f_m}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_n \end{bmatrix}. \tag{2.10}$$

2.4 Inverse Kinematic Mappings

Spatial and geometrical planning of technological operations involving manipulators is usually specified in terms of $\mathbf{x}_i \in \mathfrak{C}$ so it is important to find some or all angles $\boldsymbol{\theta}_i \in \mathfrak{C}$ to move a manipulator to $\mathbf{x}_i$. In order to do this the inverse kinematic mapping $\mathcal{F}^{-1}$ must be determined at $\mathbf{x}_i$. The inverse kinematic mapping

$$\mathcal{F}^{-1} : \mathsf{SE(3)} \rightarrow \mathsf{T}^n : \{\boldsymbol{\theta}_i\} = \mathcal{F}^{-1}(\mathbf{x}_i) \tag{2.11}$$

is under-determined for a single solution since the forward kinematic mapping $\mathcal{F}$ is many-to-one. $\mathcal{F}$ is generally not globally invertible. The topological structure of $\mathcal{F}^{-1}$ can be complicated. See §7.5.

2.7 Definition (A Critical Point Surface)
Given a forward kinematic mapping, $\mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{C}^m, m \leq n$, let $\mathbb{CP}$ denote the set of critical points of $\mathcal{F}$ which are $\boldsymbol{\theta}_i \in \mathsf{T}^n$ s.t. $\mathbf{J}$ has rank less than m. In the usual case of $\mathrm{rank}(\mathbf{J}) = m - 1$, elements of the image of $\mathbb{CP}$, $\mathcal{F}(\boldsymbol{\theta}_i)$ are called critical values. They form $m - 1$ dimensional separating surfaces called Critical Point Surfaces $\mathbb{CPS}$ (Burdick, 1988).

2.8 Definition (A Co-Regular Surface)
The set $\{\mathbb{P} = \mathsf{T}^n \backslash \mathbb{CP}\}$ are the **regular** values of $\mathcal{F}$. $\forall\ \boldsymbol{\theta}_i \in \mathsf{T}^n$, if $\exists \boldsymbol{\theta}_j \in \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{\theta}_i))$ and $\boldsymbol{\theta}_j \in \mathbb{CP}$, then $\boldsymbol{\theta}_j$ is a **co-regular point** of $\mathcal{F}$. $\mathbb{CR} = \{\boldsymbol{\theta}_j\}$ form $n - 1$ dimensional surfaces which partition T^n. Because of self-intersection, co-regular surfaces are generally not manifolds.

C^m is partitioned by $\mathbb{CPS}$, which are manifolds such that $\forall \mathbf{x}_j \in \mathbb{CPS}$, $\mathbf{J}$ loses row rank ($m \rightarrow m-1$). Equivalently, T^n is partitioned by $\mathbb{CR}$ such that $\forall \boldsymbol{\theta}_j \in \mathbb{CR}$, $\mathbf{J}$ looses row rank ($m \rightarrow m-1$). $\mathbb{CR}$ are the regions of the pre-image torus which correspond to $\mathbb{CPS}$ of the image of $\mathcal{F}$ where $\mathbf{J}$ is singular. (Burdick, 1988) proved that the maximum number of connected pre-image partitions of joint space that map onto image space by general $\mathcal{F}$ mapping to SE(3) is 16.

Pre-image partitions are usually referred to as solution branches of $\mathcal{F}^{-1}$. The number of solution branches $b_{\mathbf{x}_i}$ for a given target query $\mathbf{x}_i \in \mathfrak{C}$ is a function of $n = \dim(\mathsf{J}^n)$, of the DH parameters (see Table 2.4 taken from (Craig, 1989), pg. 119), of the physical limits of joint motion $\overline{\overline{\theta}}$, and $\mathbf{x}_i \in \mathsf{C}^m$

$$b_{\mathbf{x}_i} = f(n, \mathrm{DH}, \overline{\overline{\theta}}, \mathbf{x}_i). \tag{2.12}$$

For anthropomorphic manipulators, $\mathcal{F} : \mathsf{T}^6 \rightarrow \mathsf{C}^6$, solution branches are separated by $\mathbb{CPS} \in \mathsf{C}^6$ and $\mathbb{CR} \in \mathsf{T}^6$ into partitions where sets of direct inverse functions $\{f_1(), \ldots, f_b()\}$ can be defined

$$\{\theta_i\} = \bigcup_{j=1}^{b} f_j^{-1}(\mathbf{x}_i). \tag{2.13}$$

See (Craig, 1989; Fu et al., 1987) for derivation of sets of direct inverse functions for different classes of manipulators. In general for $\mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{C}^m$, if $n \leq m \leq 6$, a manipulator is non-redundant and some number of branches b of discrete solutions may exist $\{\theta_i\}, i = 1, \ldots, b$ for a target query $\mathbf{x}_i \in \mathfrak{C}$. For instance, a PUMA 560 (see Table 2.1) spatial manipulators can reach certain goals with 8 different configurations ([left arm × right arm] × [elbow up × elbow down] × [wrist flip × no wrist flip]).

At a $\mathbb{CPS}$, solution branches contract. At $\mathbb{CPS}$, $\mathbf{J}$ looses rank, or equivalently a manipulator loses a *dof* and certain tasks cannot be performed. In the neighbourhood of a $\mathbb{CPS}$, actions of manipulators are not well-conditioned. In general, the further a manipulator is from a $\mathbb{CPS}$ the better it is able operate isotropically, i.e. to move uniformly and apply forces uniformly in all directions. The determinant of various Jacobian expressions is typically used as a measure of isotropicality of regions of $\mathfrak{W}$. A general manipulability measure due to (Yoshikawa, 1985) is

a_i	b
$a_1 = a_3 = a_5 = 0$	≤ 4
$a_3 = a_5 = 0$	≤ 8
$a_3 = 0$	≤ 16
$\forall a_i \neq 0$	≤ 16

Table 2.4: $b = f(a_i)$ from Canonical DH.

$$M = \sqrt{|\mathbf{J}\mathbf{J}^T|}. \tag{2.14}$$

For non-redundant manipulators, this reduces to $M = |\mathbf{J}|$. At $\mathbb{CPS}$, $|\mathbf{J}| = 0$. High values of M indicate high manipulability and good manipulator designs are characterised by large sub-$\mathfrak{W}$ with high M values. Other manipulability measures have been proposed for acceleration and force isotropy. See (Craig, 1989).

2.4.1 Inverse Kinematic Trajectories

It is often necessary to determine a sequence of inverse kinematic solutions that follow a trajectory in Cartesian space, sometimes in real-time. Cartesian trajectory interpolators take two end-point Cartesian goals $(\mathbf{x}_1, \mathbf{x}_2)$, and compute q intermediate Cartesian trajectory set-points $\{\mathbf{x}_i\}$,for instance a straight line trajectory

$$\{\mathbf{x}_i\} = \frac{(\mathbf{x}_2 - \mathbf{x}_1)}{q} \times i \quad \text{for} \quad i = 1 \ldots q. \tag{2.15}$$

Subsequently an inverse set of intermediate joint angles $\{\theta_i\} = \mathcal{F}_j^{-1}(\{\mathbf{x}_i\})$ is computed on a solution branch j. Then interpolation, typically using B-splines, is used to compute smooth $\{\ddot{\theta}_i\}$ (Fu et al., 1987). The profile of $\{\ddot{\theta}_1, \ldots, \ddot{\theta}_q\}$ is usually approximately Gaussian around $\{\ddot{\theta}_{q/2}\}$, i.e. smooth acceleration from $\ddot{\theta}_1 = 0$ to max acceleration, a switch to deceleration at $\ddot{\theta}_{p/2}$, and smooth deceleration to $\ddot{\theta}_q = 0$. q is determined by the inverse dynamic servo rate. Manipulator inverse dynamic controllers

must typically be set $\ddot{\theta}_i$ goals every $20ms$ or else a manipulator stops. Fast inverse kinematics is required if during trajectory execution a new trajectory needs to be computed, for instance because sensors detect imminent obstacle collision or dynamic inversion has difficulty tracking commanded trajectories $\{\ddot{\theta}_1, \ldots, \ddot{\theta}_q\}$. During a Cartesian trajectory, if an inverse kinematic solution is required on another branch partition, a $\mathbb{CPS}$ must be traversed.

2.5 Inverse Kinematic Solutions of Non-Redundant Manipulators

Manipulator inverse kinematic mappings are generally nonlinear for every $\{\theta_i\}$ at any $\mathbf{x}_i$ due to the transcendental equations governing manipulator geometry. A manipulator is considered solvable if all joint solutions can be determined. Solving inverse kinematics for the class of all manipulators is difficult. It is particularly difficult for an under-determined equation system in the case of redundant manipulators since

> " There are no *good* general algorithms for solving systems of more than one nonlinear equation. Furthermore, it is ... very likely there *never will* be any good general methods." (Press et al., 1992), pg. 379.

Two types of nonlinearities are distinguishable in inverse kinematic mappings of non-redundant manipulators:

1. The forward mapping $\mathcal{F}$ is constructed from transcendental functions $\cos(\theta_i), \sin(\theta_i)$.
2. Multi-branch solutions. For general multi-branch nonlinear systems, if $\mathbf{J}$ is a $n \times n$ Jacobian matrix, pairwise sets of solution branches may collapse for $\mathbf{x}_i \in \mathbb{CPS}$.

Four basic methods can be distinguished to determine and represent inverse kinematic solutions of manipulators as a special case of solution of systems of nonlinear algebraic equations:

1. Closed-form analytic solution. Closed form methods manipulate algebraic or geometrical equivalences.
2. Solution group factorisation. See §7. Solution group factorisation of nonlinear algebraic equations is a special case of general methods for factorisation of nonlinear differential equations.
3. Iterative solution. Iterative inversion is generally based on differential geometry.
4. nonparametric regression on the solution where input and output distributions are initially unknown:
   (a) a direct inverse representation of a solution,
   (b) an indirect inverse representation of a solution.

2.5.1 Closed-Form

Geometric and algebraic solution methods can be used to derive
closed form analytic expressions of inverse kinematics of some classes of manipulators. Geometrical solution methods determine substitutions in order to derive solutions. Algebraic solution methods for solving transcendental equations are generally based on transformation of transcendental equations

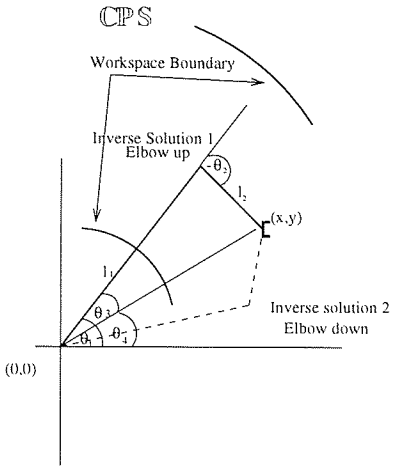

Figure 2.3: The Geometric Solution of Planar 2-**R**.

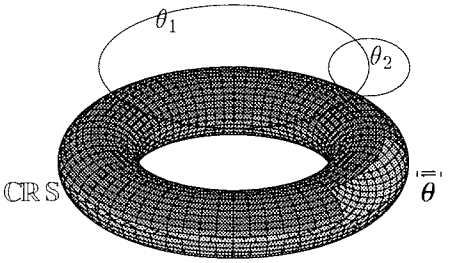

Figure 2.4: The T^2 image of $\mathcal{F}^{-1}$. The light shading represents a Co-Regular Surface and a region of Joint Space where the inverse mapping does not exist because of joint limits.

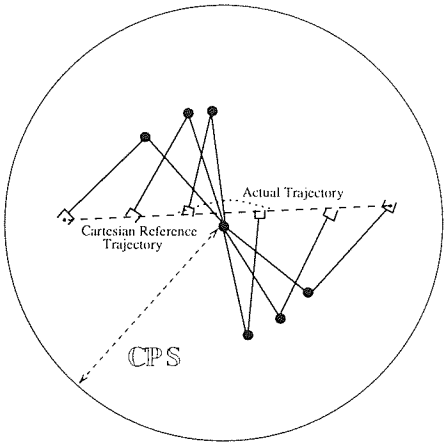

Figure 2.5: A Cartesian Trajectory Failure near the Origin Singularity of Planar 2-**R**. $|\mathbf{J}| \rightarrow 0; \theta_2 = \pi$.

to equations of polynomials of degree 4 or less (Craig, 1989; Fu et al., 1987). (Pieper, 1968) derived a general solution for spatial manipulators with three intersecting joint axes (i.e., a 3**R** wrist with wrist offset parameters all zero) based on finding the roots of fourth degree polynomials constructed on transformations of $\boldsymbol{\theta}_i$. Solutions for 6 equations in 6 unknowns are derived by inspection of sub-equations in the homogeneous transform system of a forward mapping $\mathcal{F}$ in (2.5). See (Craig, 1989), pps. 128 - 141, for a derivation of the analytical solution of the PUMA 560. Availability of a closed form solution is the primary reason that many industrial manipulators, for instance the PUMA[5] 560, have a spherical wrist with no offsets.

Geometric and algebraic solutions of planar 2-**R** (see Figure 2.3) kinematics

$$\mathcal{F} : \mathsf{T}^2 \rightarrow \mathfrak{W} \subset \mathbb{R}^2 : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i) \tag{2.16}$$

are derived in Appendix D as a preliminary to later discussion of inverse kinematic solution topologies. There are two $\mathbb{CP}\mathbb{S}$ and two solution branches of the inverse kinematic mapping. There are two possible configurations of planar 2-**R** at all points in $\mathfrak{W}$ except at $\mathbb{CP}\mathbb{S}$ where there is 1 (the envelope) or ∞ (the origin). The sign of θ_2 determines the sign of k_2 in Appendix D which determines one of two possible θ_1 solutions:

1. If $\theta_2 = \pi$, $x = y = l_1 - l_2$ in (D.22), $\theta_1 \in [0, 2\pi)$. If $l_1 = l_2$, the $\mathbb{CP}\mathbb{S}$ collapses into a singularity point at the origin. See Figure 2.5.

[5]The PUMA 560 has undesirable mechanical properties such as a fragile wrist and reduced payload capacity. Manipulators with wrist offsets such as the Robotics Research K-1207 have better mechanical properties. See §8.3.

2. At $\theta_2 = 0$, planar 2-**R** is fully extended touching $\mathfrak{W}$ boundary. There is a single solution for θ_1. $\mathfrak{W}$ boundaries are $\mathbb{CPS}$.

$\mathbb{CPS}$ are the only place where planar 2-**R** can change solution configuration during execution of Cartesian trajectories in the base space $\mathfrak{W}$. In Figure 2.5, with $l_1 = l_2$, these correspond to switching from elbow up to down solutions. Figure 2.4 is a representation of the T^2 pre-image of $\mathcal{F}$ of planar 2-**R** with some arbitrary joint limits and $\mathbb{CPS}$ indicated.

In general, a $\mathbb{R}^2 \supset \mathfrak{W} \rightarrow \mathsf{T}^2$ trajectory inversion encounters problems as the $\mathbb{CPS}$ are approached. Figure 2.5 shows a path deviation as the origin singularity is approached. The desired trajectory requires constant linear Cartesian velocity $\dot{\mathbf{x}} = a$. As a singularity is approached, one or more $\dot{\theta}$ are required to increase beyond a mechanical upper bound on angular velocity. Since this is impossible, a manipulator will deviate from a specified path which may cause collisions. Efficient algorithmic representations of analytical solutions can be pipelined and parallelised and before compilation on special purpose real-time hardware (Lee, 1991a; Zhang and Paul, 1991).

2.5.2 Differential Iteration

If a closed form inverse solution is unknown, a basic method to robustly generate an inverse solution for a manipulator is iterative[6] differential inversion. This requires either a forward model or sensor measurements $\mathbf{x}_t$ of the end effector Cartesian position. Many iterative solution methods minimise a Jacobian-based error function of position error $\mathbf{x}_t - \mathbf{x}_g$. Provided Newton-Raphson iteration or its more effecients variants are initialised at $\boldsymbol{\theta}_t$ in a convex region of $\mathcal{F}$ containing $\mathbf{x}_g$, then iteration of

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{J}^{-1}(\mathbf{x}_t - \mathbf{x}_g), \tag{2.17}$$

where α is determined by a line search, and the update is $\mathbf{x}_{t+1} = \mathcal{F}(\boldsymbol{\theta}_{t+1})$ results in $\mathbf{x}_t - \mathbf{x}_g \rightarrow 0$ as $t \rightarrow \infty$. The problems with Jacobian methods applied to multi-branch inverse mappings are:

❶ the solution branch b reached depends on initial configuration $\boldsymbol{\theta}_t$ and thus may be locally but not globally optimal.

❷ when $\text{rank}(\mathbf{J}) \rightarrow n - 1$, there are singularities that cause numerical inversion difficulties.

❷ can be solved by generalising matrix inversion to $m \times n$ matrices (Samson et al., 1991). If $\mathbf{J}$ is a $n \times n$ matrix that looses rank, this converts it to an $m \times n$ matrix where $m = n - 1$. If $\mathbf{A}$ is an $m \times n$ matrix of full rank $m < n$, then

2.9 Definition (A Generalised Matrix Inverse)
$\mathbf{A}^g$ is a generalised inverse $\iff \mathbf{A} = \mathbf{A}\mathbf{A}^g\mathbf{A}$.

2.10 Definition (A Reflexive Generalised Matrix Inverse)
$\mathbf{A}^r$ is a reflexive generalised inverse $\iff$ $\mathbf{A}$ is a generalised inverse and $\mathbf{A}^r = \mathbf{A}^r\mathbf{A}\mathbf{A}^r$.

2.11 Definition (A Matrix Pseudoinverse)
$\mathbf{A}^+$ is a unique pseudoinverse $\iff$ it is a generalised reflexive inverse and

$$(\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A}, \tag{2.18}$$

$$(\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+, \tag{2.19}$$

$$\mathbf{A}^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}. \tag{2.20}$$

[6] This is also known as closed-loop or feedback inversion in control theory.

Replacing $\mathbf{J}^{-1}$ by $\mathbf{J}^{+}$ in (2.17), a minimum norm inverse solution $\Delta\hat{\theta}_t$ is defined

$$\Delta\hat{\theta}_i = \mathbf{J}^{+}(\mathbf{x}_t - \mathbf{x}_g), \tag{2.21}$$

$$||\mathbf{J}\Delta\hat{\theta}_t - (\mathbf{x}_t - \mathbf{x}_g)|| = \min_{\Delta\theta}||\mathbf{J}\Delta\theta - (\mathbf{x}_t - \mathbf{x}_g)||. \tag{2.22}$$

While this deals with ❷, solution cyclicity is not guaranteed. Pseudoinverse tracking does not always avoid $\mathbb{CPS}$. This may cause violation of geometrical path planning constraints if a manipulator switches configuration branches. Pseudoinversion may not be tractable for real-time trajectory inversion. Real-time differential trajectory inversion require reduction of $\mathbf{J}$ evaluation and inversion to a minimal number of operations by recursive representation implemented on application specific hardware (Lee, 1991a; Zhang and Paul, 1991).

Good iterative solution methods are local and cyclic (Baker, 1990).

2.12 Definition (A Local Inverse Kinematic Solution Method)
A local solution method can determine a solution from the current $\mathbf{x}_t, \theta_t$ and goal end effector position $\mathbf{x}_g$.

2.13 Definition (A Cyclic Inverse Kinematic Solution Method)
A solution method is cyclic if for a closed path in $\mathfrak{M}$, the method generates a closed solution path in joint space. See §7.3.

2.5.3 Eigendecomposition of Matrix Polynomials

(Manocha and Canny, 1994) developed an efficient and stable iterative method to compute the set of inverse kinematic solutions of any spatial or dextrous serial revolute joint manipulators. The (Manocha and Canny, 1994) solution is based on transformation of the solution of the forward kinematic transform system into a problem of manipulating matrix polynomials, followed by an eigendcomposition computation, and back-substitution of the solution.

To set up the eigendecompostion (2.5) is rearranged to

$${}^{2}_{3}\mathbf{T}{}^{3}_{4}\mathbf{T}{}^{4}_{5}\mathbf{T} = {}^{0}_{1}\mathbf{T}{}^{1}_{2}\mathbf{T}{}^{0}_{n}\mathbf{T}{}^{n-1}_{n}\mathbf{T} \tag{2.23}$$

The entries of the left hand side matrix of (2.23) are functions of $\theta_3, \theta_4, \theta_5$. The entries of the right hand side matrix of (2.23) are functions of $\theta_1, \theta_2, \theta_6$. After simplification, variable elimination, substitution and rearrangement of (2.23), it is possible to determine $\mathbf{A}\mathbf{b} = 0$ where $\mathbf{A}$ is a 12×12 matrix whose elements are quadratic polynomials in $\tan(\frac{\theta_3}{2})$ and $\mathbf{b}$ is a 12×1 vector of power products of $\tan(\frac{\theta_4}{2}), \tan(\frac{\theta_5}{2})$. If $\mathbf{A}$ is singular, i.e. $\det(\mathbf{A}) = 0$, $\mathbf{A}$ has a non-trivial kernel $\mathbf{b}$, i.e. non-zero solutions for $\tan(\frac{\theta_4}{2}), \tan(\frac{\theta_5}{2})$. Algorithms to find the roots of

$$\det(\mathbf{A}) = 0, \tag{2.24}$$

a polynomial of degree 24 in $\tan(\frac{\theta_3}{2})$, are inefficient and numerically unstable. Hence (Manocha and Canny, 1994) reduce the problem of computing the roots of $\det(\mathbf{A})$ to a problem of computing corresponding eigenvalues of a transformed eigendecompostion of matrix polynomials. . A matrix polynomial is a matrix valued function defined by

$$\mathbf{L}(x) = \sum_{i=0}^{k} \mathbf{A}_i x^i. \tag{2.25}$$

(2.25) is a monic polynomial if $\mathbf{A}_k = \mathbf{I}$. Since the elements of $\mathbf{A}$ contain quadratic functions of $\tan(\frac{\theta_3}{2})$, $\mathbf{A}\mathbf{b} = 0$ can be transformed to a matrix polynomial

$$\mathbf{L}(\tan(\frac{\theta_3}{2})) = \mathbf{A}_2 \tan^2(\frac{\theta_3}{2}) + \mathbf{A}_1 \tan(\frac{\theta_3}{2}) + \mathbf{A}_0, \tag{2.26}$$

where $\mathbf{A}_i$ are 12×12 numerical matrices. If $\mathbf{A}_2$ is well-conditioned, a monic polynomial transformation of (2.26) is defined by

$$\bar{\mathbf{L}}(\tan(\frac{\theta_3}{2})) = \mathbf{I} \tan^2(\frac{\theta_3}{2}) + \mathbf{A}_2^{-1}\mathbf{A}_1 \tan(\frac{\theta_3}{2}) + \mathbf{A}_2^{-1}\mathbf{A}_0. \tag{2.27}$$

Since the determinant of $\bar{\mathbf{L}}$ has the same roots as the determinant of $\mathbf{L}$ which in turn has the same roots as the determinant of $\mathbf{A}$ in (2.24), θ_3^0 a root of

$$\det(\bar{\mathbf{L}}(\tan(\frac{\theta_3}{2}))) = 0, \tag{2.28}$$

(2.28) is also a root of (2.24). (Manocha and Canny, 1994) prove that given a matrix polynomial $\bar{\mathbf{L}}(x)$, the roots of the polynomial corresponding to its determinant are the eigenvalues of the matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_m \\ -\bar{\mathbf{A}_0} & -\bar{\mathbf{A}_1} & -\bar{\mathbf{A}_2} & \dots & -\bar{\mathbf{A}_k - 1} \end{bmatrix} \tag{2.29}$$

where $\bar{\mathbf{A}_i} = \mathbf{A}_k^{-1}\mathbf{A}_i$, which for the kinematic problem becomes

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{A}_2^{-1}\mathbf{A}_0 & -\mathbf{A}_2^{-1}\mathbf{A}_1 \end{bmatrix} \tag{2.30}$$

Once the eigenvalues of (2.30) are determined, it is possible to determine corresponding eigenvectors, and after some substitutions $\{\theta_3^0\}$. Subsequently θ_4, θ_5 are determined and finally all solutions by back-substitution. (Manocha and Canny, 1994) extend the basic transformation of an evaluation of the roots of a determinant to an eigendecomposition of matrix polynomials to handle ill-conditioned steps in the algorithm, e.g. when $\mathbf{A}_2$ is singular or close to singular in (2.27). (Manocha and Canny, 1994) tested the solution algorithm on a variety of manipulator inversion problems and report an average algorithm execution time of 11 milliseconds on IBM RS\6000 workstation. This is fast enough for computation of inverse Cartesian trajectories.

2.5.4 Nonparametric Approximation

If a manipulator is not parametrically solvable, nonparametric nonlinear function approximation of inverse kinematics is also possible, e.g. (Miller, 1989; Miller et al., 1990; Kuperstein and Rubinstein, 1989; Cooperstock and Milios, 1993; Mel, 1990; Ritter et al., 1992). See §5.3. Generalised linear adaptive control theory (see (Sontag, 1990)) provides a general framework for definition of possibilities for nonparametric inversion of nonlinear[7] systems. Let $\{\mathbf{x}_t, \dot{\mathbf{x}}_t\}$ be a state space, $\mathbf{u}_t$ be control inputs,

[7]Nonlinear control theory generalises linear control notions of reachability, observability, and stability of $\dot{\mathbf{x}}_t = f(\mathbf{x}_t, \mathbf{u}_t)$, where $\mathbf{x}_t, \dot{\mathbf{x}}_t$ is a state space and $\mathbf{u}_t$ are control inputs. If a state space manifold $\mathbb{M}$ is appropriately defined, state space evolution over t without inputs is defined by a drift flow D on $\mathbb{M}$ and time independent inputs define a control flow C on $\mathbb{M}$. The Lie bracket operator $[C, D]$ defines the evolution of the system with control inputs on the manifold $f : \mathbb{M} \times \mathbb{M} \rightarrow \mathbb{TM} : E = [C, D]$ where $\mathbb{TM}$ is the tangent fibre bundle manifold. If control inputs C are also dependent on time, i.e. C_t, $f_t : \mathbb{M} \times \mathbb{M}_t \rightarrow \mathbb{TM} : E = [C, D_t]$ is defined on collections of trivial fibre bundles, i.e. a fibre bundle state space is defined. See §7.

and $\mathbf{o}_t$ observable system outputs related by the following system equations

$$\mathbf{x}_{t+1} = f_1(\mathbf{x}_t, \dot{\mathbf{x}}_t, \mathbf{u}_t), \tag{2.31}$$

$$\mathbf{o}_{t+1} = f_2(\mathbf{x}_t, \dot{\mathbf{x}}_t, \mathbf{u}_t). \tag{2.32}$$

A nonlinear state space defined on a manifold $\mathbf{x}_t, \dot{\mathbf{x}}_t$ may be approximated by a collection of first order Taylor expansions. Piecewise linear system identification F involves discovering a good partition of $\mathbf{x}_t, \dot{\mathbf{x}}_t$ and a set of inverse models of $\mathbf{x}_t, \dot{\mathbf{x}}_t$ for use in control applications (Ljung, 1987). Transition between Taylor approximation of state space partitions and associated inverse models is discontinuous so methods are required to smooth control. In a general linear feedback control system, after a control state space is sectioned into quasi-linear elements, each with an associated linear inverse, observability, reachability, stability and robustness are provable (Sontag, 1990).

2.14 Definition (System Observability)

A controlled system $\{\mathbf{x}_t, \dot{\mathbf{x}}_t, \mathbf{u}_t\}$ is completely observable if observation of $(\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \ldots, \mathbf{o}_n)$, a sequence of outputs, and controls $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots, \mathbf{u}_n)$ is sufficient to identify an initial system state $(\mathbf{x}_0, \dot{\mathbf{x}}_0)$.

2.15 Definition (System Reachability)

A controlled system is completely reachable if for any state $\mathbf{x}_i, \dot{\mathbf{x}}_i$, there exists an initial state, $\mathbf{x}_0, \dot{\mathbf{x}}_0$, and an input sequence $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ such that $\mathbf{x}_i = \mathbf{x}_n, \dot{\mathbf{x}}_i = \dot{\mathbf{x}}_n$.

2.16 Definition (System Stability)

A system in a steady state $(\mathbf{x}_0, \dot{\mathbf{x}}_0)$ is stable if for each neighbourhood V of $(\mathbf{x}_0, \dot{\mathbf{x}}_0), \exists \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ such that $\mathbf{x}_0 = \mathbf{x}_n, \dot{\mathbf{x}}_0 = \dot{\mathbf{x}}_n$.

2.17 Definition (System Robustness)

A control system is robust if control is guaranteed even if the assumed model F of the system f_1 to be controlled is incorrect (with the allowed deviations quantified in appropriate norms) or under the possibility of imperfect controller design.

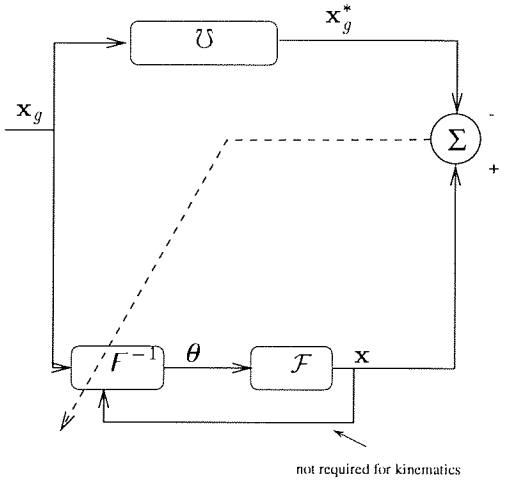

Figure 2.6: Direct Inversion.

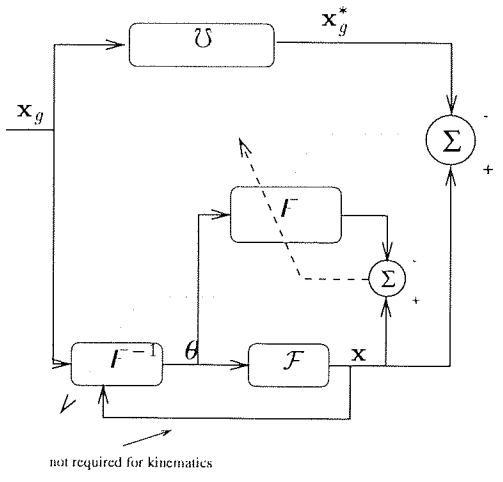

Figure 2.7: Indirect Inversion.

There are two principal ways to calibrate nonparametric models - direct calibration and indirect calibration. In calibration of direct inverse nonparametric models, $\mathcal{F}$ is controlled by a signal $\boldsymbol{\theta}_i$ and outputs signal $\mathbf{x}_i$. See Figure 2.6. A reference signal is a signal $\mathbf{x}_g$ which is input with $\mathbf{x}_i$ to a controller F^{-1}. $\boldsymbol{\theta}_i = F^{-1}(\mathbf{x}_g, \mathbf{x}_i)$.

If $\mathbf{x}_g$ is not in the same coordinate system as $\mathbf{x}_i$, $\mathbf{x}_g^*$ is derived from $\mathbf{x}_g$ by means of a reference model $\mho$ (See Figure 2.6). $\mho$ translates $\mathbf{x}_g$ to $\mathbf{x}_g^*$, the range of $\mathcal{F}$ outputs $\mathbf{x}_i$, such that the two can be combined in a single error signal $\mathbf{x}_i - \mathbf{x}_g^*$. $\mathbf{x}_i - \mathbf{x}_g^*$ is subsequently used to correct F^{-1}. The adaptation task for F^{-1} is to minimise $\| \mathbf{x}_i - \mathbf{x}_g^* \|$ where $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$.

An error signal $\mathbf{x}_i - \mathbf{x}_g^*$ is information on how $\mathcal{F}$ must change, and not $\boldsymbol{\theta}_i$, the output of F^{-1}. In order to correct F^{-1}, an inverse $\mathcal{F}^{-1}$ is required. If $\mathcal{F}^{-1}$ is not available, (and typically it is not, since if it was, it could itself be used for F^{-1}), a forward model F is required to evaluate $\frac{\partial \mathbf{x}_i}{\partial \theta_i^j}$, the derivative of the components of a vector $\mathbf{x}_i$ with respect to the j components of $\boldsymbol{\theta}_i$.

In indirect calibration, a direct model must first be calibrated. Determination of F's order and parameters is known as system identification. Forward kinematic model order identification is trivial. It is the DH parametrised forward transformation of §2.3.

After F is identified, given that $\boldsymbol{\theta}_i = F^{-1}(\mathbf{x}_g, \mathbf{x}_i)$ are input to system $\mathcal{F}$ which is closely approximated by F, F^{-1} can be identified by minimising $\| \mathbf{x}_i - \mathbf{x}_g^* \|$ by the chain rule. With a SSE

$$E = \frac{1}{2} \sum_i (\mathbf{x}_i - \mathbf{x}_g^*)^2, \tag{2.33}$$

and an assumption that neither F nor the reference signal model $\mathbf{x}_g^*$ are modifiable, E can be minimised by a gradient update of w_k, the parameters of F^{-1}, by

$$\frac{\partial E}{\partial w_k} = \sum_i (\mathbf{x}_i - \mathbf{x}_g^*) \frac{\partial \mathbf{x}_i}{\partial w_k}, \tag{2.34}$$

$$\frac{\partial \mathbf{x}_i}{\partial w_k} = \sum_j \frac{\partial F(\boldsymbol{\theta}_i)}{\partial \theta_i^j} \frac{\partial \theta_i^j}{\partial w_k}, \tag{2.35}$$

where j indexes components of $\boldsymbol{\theta}_i$ for F^{-1} evaluated at $\mathbf{x}_g, \mathbf{x}$. Kinematics of manipulators is not strictly (but see §2.5.5) dependent on previous states. In this special case $\boldsymbol{\theta}_i = F^{-1}(\mathbf{x}_g)$ only. See Figure 2.7.

2.5.5 Semiparametric Approximation

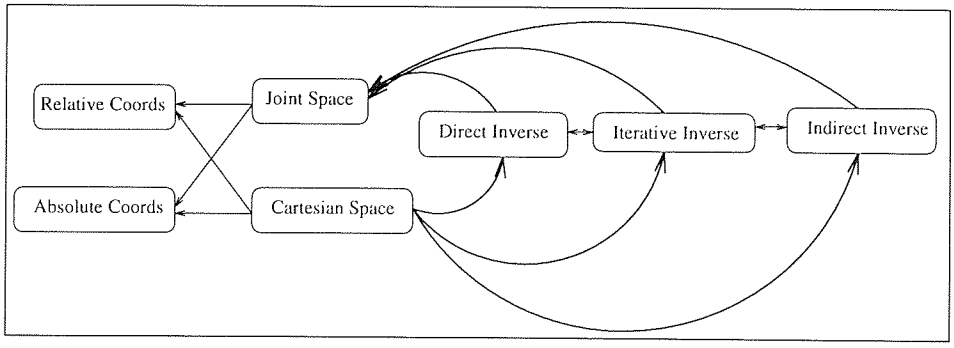

Figure 2.8: Nonparametric and Iterative Inverse Kinematic Solution Methods for Non-Redundant Manipulators. Arrows indicate Possible Combinations of Solution Methods and Coordinate Systems.

An incorrectly parameterised analytical direct inverse or forward model may be combined with nonparametric forward modelling and inversion. Correction of incorrectly parameterised analytical inverses, a form of semiparametric regression, may involve (see Table 2.5, pg. 48)

1. Correction of $\mathbf{x}_i \in \mathfrak{C}^n$ or of $\boldsymbol{\theta}_i \in \mathfrak{J}^m$.

2. absolute $\mathbf{x}_i, \boldsymbol{\theta}_i$ or relative $\Delta\mathbf{x}_i, \Delta\boldsymbol{\theta}_i$ coordinate correction.

Both direct and indirect inversion may be supplemented by iterative correction. In an indirect case when a forward model is learnt, iteration through the forward model and the indirect inverse approximation may also be performed to increase inverse mapping accuracy. This corresponds to a feedback phase with constant reference signal in terms of Figure 2.6. Figure 2.8 is a schematic outline of possible interactions between representations of input and output spaces of solutions and methods and representations of the inverse kinematics of $\mathbf{x}_i$.

Given a Cartesian goal position $\mathbf{x}_i$, the four ways that a nonparametric model can correct an incorrectly parameterised analytical inverse mapping $\mathcal{F}_a^{-1}$ to find correct $\boldsymbol{\theta}_i$ are shown in Table 2.5. It is possible to alter the inputs to $\mathcal{F}^{-1}$ either in absolute $\mathbf{x}_i$ coordinates or relative (to $\mathbf{x}_i$) coordinates to generate a correct $\mathcal{F}^{-1}$. This corresponds to a remap of the input space that makes the analytical inverse function correct. It is also possible to correct an analytical function by altering the outputs again in either absolute or relative coordinates.

Coordinates	$\mathfrak{C}$ Cartesian Space	$\mathfrak{J}$ Joint Space
Absolute	$\boldsymbol{\theta}_i = \mathcal{F}_a^{-1}(\mathcal{F}(\mathbf{x}_i))$	$\boldsymbol{\theta}_i = \mathcal{F}(\mathcal{F}_a^{-1}(\mathbf{x}_i))$
Relative	$\boldsymbol{\theta}_i = \mathcal{F}_a^{-1}(\mathcal{F}(\mathbf{x}_i) + \mathbf{x}_i)$	$\boldsymbol{\theta}_i = \mathcal{F}_a^{-1}(\mathbf{x}_i) + \mathcal{F}(\mathbf{x}_i)$

Table 2.5: Four Modes in which a Nonparametric Model $\mathcal{F}$ can Correct an Analytical Parametric Model $\mathcal{F}_a^{-1}$ of Inverse Kinematics.

Whether correction of an analytical inverse function by pre-processing its input space or post-processing its output space is more appropriate than re-calibration of equation parameters depends on the type of nonparametric model being used and data collection constraints. Semiparametric modelling of a nonlinearity with an analytic inverse function $\mathcal{F}^{-1}$ corrected by a nonparametric component $\mathcal{F}$ is often used in adaptive control theory, for instance $\boldsymbol{\theta}_i = \mathcal{F}^{-1}(\mathbf{x}_i) + \mathcal{F}(\mathbf{x}_i)$. An incorrectly parameterised analytical inverse kinematic solution $\mathcal{F}_a^{-1}$ can be corrected by a nonparametric model fitted to sensor measurements of inverse kinematic errors in position and orientation. This may be desirable if it is not possible to get into or exchange modules of commercial controllers to change DH parameters even if they can be re-measured correctly. Since commercial black box analytic inverters model primary nonlinearities, semiparametric correction requires approximation of less nonlinearity than full inverse approximation. For some commercial inverse kinematic software, only $\mathfrak{C}$ goals $\mathbf{x}_i$ may be specified, especially for trajectory inversion, so inverse kinematics must be corrected at the $\mathfrak{C}$ interface. In this case, kinematic commands must be issued to $\mathcal{F}$ in $\mathfrak{C}$ coordinates $\mathbf{x}_i$. Suppose that $\boldsymbol{\theta}_i^a = \mathcal{F}_a^{-1}(\mathbf{x}_i)$ is an incorrect analytical inverse leading to $\mathbf{x}_i^a = \mathcal{F}(\boldsymbol{\theta}_i^a)$, i.e. $\mathbf{x}_i^a = \mathcal{F}(\mathcal{F}_a^{-1}(\mathbf{x}_i))$. The parameters of a nonparametric $\mathcal{F}^{-1}$ to reach $\mathbf{x}_i = \mathcal{F}(\mathcal{F}^{-1}(\mathbf{x}_i^a))$ can be determined by an indirect inverse equation

$$\mathbf{x}_i = \mathcal{F}(\mathcal{F}^{-1}\left(\mathcal{F}(\mathcal{F}_a^{-1}(\mathbf{x}_i))\right)). \tag{2.36}$$

If an approximation $\mathcal{F}$ of $\mathcal{F}$ has been identified for a data set $\{\boldsymbol{\theta}_i, \mathbf{x}_i\}, i = 1, \ldots, p$ by various methods for instance update of incorrect forward equations $\mathcal{F}_a$ at the joint coordinate system interface, i.e.

$$\min_{\mathcal{F}} \left(\mathcal{F}(\boldsymbol{\theta}_i) - \mathbf{x}_i\right)^2, \tag{2.37}$$

where

$$\mathcal{F}(\boldsymbol{\theta}_i) = \mathcal{F}(\boldsymbol{\theta}_i) + \Delta(\boldsymbol{\theta}_i), \tag{2.38}$$

then minimisation of an error, for instance SSE, function

$$\min_{F^{-1}} \left[\sum_{i=1}^{p} (FF^{-1}\left(FF_a^{-1}(\mathbf{x}_i)\right) - \mathbf{x}_i)^2 \right] \tag{2.39}$$

determines F^{-1}. If there are no interface update requirements, (2.39) reduces to basic indirect inversion. It also possible to combine iterative approximation with semiparametric approximation. While feedback from $\mathbf{x}_i$ to F^{-1} in Figure 2.7 is not strictly required for indirect inversion of manipulator kinematics, it is mentioned since given an incorrect controller, and a constant reference signal, iterative feedback may be employed to generate a correct control action. Iterative feedback methods are often employed for on-line correction of erroneously parameterised nominal closed form solutions if visual or other sensor measurements of manipulator position are available. See (Samson et al., 1991).

2.6 Inverse Kinematic Solutions of Redundant Manipulators

Dextrous manipulators have applications[8] in hazardous environments, for instance spent nuclear fuel re-processing and low earth orbital satellite deployment, repair and assembly. Dextrous manipulators require secondary optimisation, for instance for path planning collision avoidance, to determine an inverse kinematic solution criteria. Dextrous manipulators can

1. avoid trajectory singularities common with non-redundant manipulators. See Figure 2.5.
2. maximise dynamic variation over $\mathfrak{W}$.
3. maximise manipulability M (see §2.4) while operating in restricted or cluttered $\mathfrak{W}$.
4. increase probability of robust manipulator operation in freezing or scorching conditions where joints may catastrophically freeze or jam and repair is infeasible.
5. optimise path planning kino-dynamic constraints such as minimisation of energy or acceleration jerk.

An obstacle can be a solid object or a forbidden area (e.g. $\mathbb{CP}^{\mathbb{S}}$) near where the robot can not function well. For coarse geometrical path planning, let obstacles be represented by bounding convex hulls $\{\mathbf{c}, \mathbf{r}_i\}$ having a centroid $\mathbf{c}$ and radii $\mathbf{r}_i$ to faces. For collision avoidance, given an initial position, S and final position G and a set of stationary obstacles $\{\mathbf{c}, \mathbf{r}_i\}$ projected to Free-space as $\{\mathcal{Z}_i\}$, the path planning problem is to find $\boldsymbol{\theta}(t)$ a continuous function of time $[t_s, t_g]$ specifying a configuration of the robot arm such that $\boldsymbol{\theta}(t_s) = S, \boldsymbol{\theta}(t_g) = G$, and $\boldsymbol{\theta}(t) \bigcap \mathcal{Z}_i = \emptyset$ for t in $[t_s, t_g]$.

$\mathbf{J}$ of dextrous manipulators are non-square, thus singular in most of $\mathfrak{W}$. This corresponds to an under-determined solution space or infinitely many solutions in most of $\mathfrak{W}$.

2.1 Theorem (Topology of Inverse Solution of Dextrous Manipulator (Burdick))
The topology of the pre-image of a regular point $\mathbf{x}_i \in \mathfrak{W}$ of dextrous manipulators with unlimited $\mathbf{R}$ is the group product $\mathbb{C}_b \times \mathsf{T}^{n-m}$ (Burdick, 1988).

(Burdick, 1988) proved that $\bigcup_{i=1}^{16} \mathsf{T}_i^{n-m}$ is the topological structure of the complete solution set for $\mathbf{x}_i$ inverse kinematic query of any all-$\mathbf{R}$ manipulator operating in SE(3). In general the pre-image of a particular $\mathbf{x}_i$ consists of a finite number b of distinct solution branches, i.e. it is a cyclic group $\mathbb{C}_b$. Each

[8] The cost of high precision dextrous manipulators, \$0.5m+, is due to the large cost of R+D of hardware and software for small markets, of the order of tens of units.

of the inverse branches is a torus T^{n-m} of dimension $n-m$. Figure 2.9 presents a visualisation of the inverse solution topology of dextrous rotational joint manipulators. The following solution methods may be used for computing solutions on inverse solution T^{n-m} manifolds of redundant manipulators

1. regularised nonparametric (see §5.3.2),
2. Jacobian augmentation,
3. optimisation,
4. Lyapunov stabilisation,
5. composition of direct inverse functions.

The last four methods are discussed in §2.6.1 - §2.6.5.

2.6.1 Augmented Jacobian

Iterative inversion is the most widely used method to generate solutions for dextrous manipulators. $\mathbf{J}^+$, the pseudoinverse of a Jacobian may be used to generate an inverse solution. Basic iterative Jacobian pseudoinversion may be combined with

1. partial analytical inverse and reduced Jacobian pseudoinversion (Kircanski, 1993),
2. additional task constraint functions (Kreutz-Delgado et al., 1992) and augmented Jacobian inversion.

If $n-m$ additional constraint functions are introduced directly in task space or by adding $n-m$ rows $f_{m+1}^1, \ldots, f_n^n$ to the non-square Jacobian

$$\mathbf{J}^\Diamond = \begin{bmatrix} & \mathbf{J} & \\ f_{m+1}^1() & \cdots & f_{m+1}^n() \\ \vdots & \ddots & \vdots \\ f_n^1 & \cdots & f_n^n() \end{bmatrix}. \tag{2.40}$$

then a trajectory may be followed defined by pseudoinverse, optimisation or Lyapunov methods on the resulting augmented Jacobian $\mathbf{J}^\Diamond$.

2.6.2 Optimisation

Global optimisation methods may be used to select solutions on redundant manifolds (Nakamura, 1991). In trajectory optimisation for example, side constraints are introduced to maximise manipulability or distance from singularities. Given a current Cartesian position $\mathbf{x}_{t_0} = \mathcal{F}(\boldsymbol{\theta}_{t_0})$ and a goal position $\mathbf{x}_{t_g}$, optimisation determines a trajectory $\{\boldsymbol{\theta}_t\}$ that minimises a cost integral

$$\int_{t_0}^{t_g} \left(\frac{1}{2} \dot{\boldsymbol{\theta}_t}^T \mathbf{W}^{-1} \dot{\boldsymbol{\theta}_t} + M_t \right) dt, \tag{2.41}$$

under the kinematic constraint that $\mathbf{x}_t = \mathcal{F}(\boldsymbol{\theta}_t)$. $\mathbf{W}$ is an application specific symmetric positive definite[9] weighting matrix. M_t is a function side constraint, for instance $M_t = \sqrt{|\mathbf{J}\mathbf{J}^T|}$ (c.f. (2.14)). This solves the inverse kinematic problem since $\mathbf{x}_{t_g} = \mathcal{F}(\boldsymbol{\theta}_{t_g})$.

[9] A positive definite matrix $\mathbf{W}$ has real positive eigenvalues $\nu \in \mathbb{R}^+, \forall \dot{\boldsymbol{\theta}} \in \mathbb{R}^n, \quad \dot{\boldsymbol{\theta}}^T \mathbf{W} \dot{\boldsymbol{\theta}} > 0$.

2.6.3 Lyapunov Function Minimisation

For a given Cartesian goal position $\mathbf{x}_g$ and a current position $\mathbf{x}_t = \mathcal{F}(\boldsymbol{\theta}_t)$, positioning error is defined

$$\mathbf{x}_e = \mathbf{x}_t - \mathbf{x}_g = \mathcal{F}(\boldsymbol{\theta}_t) - \mathbf{x}_g. \tag{2.42}$$

An instance of a Lyapunov function L is a scalar function

$$L = \frac{1}{2}\mathbf{x}_e^T \mathbf{A} \mathbf{x}_e, \tag{2.43}$$

where $\mathbf{A}$ is positive definite. The first derivative of L is

$$\begin{aligned} \dot{L} &= \mathbf{x}_e^T \mathbf{A} \dot{\mathbf{x}_t} && (2.44)\\ &= \mathbf{x}_e^T \mathbf{A} \mathbf{J} \dot{\boldsymbol{\theta}_t}. && (2.45) \end{aligned}$$

A gradient update of $\boldsymbol{\theta}_t$ by

$$\Delta\boldsymbol{\theta} = -\mathbf{J}^T \mathbf{A}^T \mathbf{x}_e \tag{2.46}$$

causes an update on L such that

$$\begin{aligned} \Delta L &= -(\mathbf{J}^T \mathbf{A}^T \mathbf{x}_e)^T (\mathbf{J}^T \mathbf{A}^T \mathbf{x}_e) && (2.47)\\ &= -\Delta\boldsymbol{\theta}^T \Delta\boldsymbol{\theta} && (2.48)\\ &\leq 0. && (2.49) \end{aligned}$$

Asymptotically $L \rightarrow 0$ as $t \rightarrow \infty$. $L = 0$ if $\Delta\boldsymbol{\theta} = 0$.

$$\Delta\boldsymbol{\theta} = -\mathbf{J}^T \mathbf{A}^T \mathbf{x}_e = 0 \begin{cases} \text{if} \quad \mathbf{x}_e = 0 \quad \text{(desired position)}, \\ \text{if} \quad \mathbf{A}^T \mathbf{x}_e \in \ker(\mathbf{J}^T). \end{cases} \tag{2.50}$$

In the latter case, $\mathbf{J}^T$ is singular and $\mathbf{x}_e \neq 0$ so $\mathbf{x}_g$ is not reached.

2.6.4 Direct Inverse Functions

A direct inverse function representation of the inverse mapping $\mathcal{F}^{-1}$ of a redundant manipulator by introduction of parameterisations of redundancy was proposed by (Wampler, 1988; Wampler, 1988; DeMers, 1993). A direct inverse function f_b is defined by

$$f_b : \mathsf{C}^m, \mathsf{T}^{n-m} \rightarrow \mathsf{T}^n : \boldsymbol{\theta}_i = f_b(\mathbf{x}_i, \mathbf{p}) \tag{2.51}$$

where $b = 1, \ldots, z$, indexes solution branches and z is the upper bound on the number of solution branches. A solution branch manifold is typically a sub-manifold of the torus T_i^{n-m}. $\mathbf{p}$ is a parameterisation of T_i^{n-m}.

2.6.5 A Comparison of Solution Methods

A general disadvantage of optimisation, augmented Jacobian, and Lyapunov methods is that they are computationally intensive. A particular disadvantage of iterative inversion of $\mathbf{J}^{\Diamond +}$ is that while it is locally cyclic, i.e. for trajectories that do not cross $\mathbb{CPS}$, $\mathbf{J}^{\Diamond +}$ may be singular at configurations where

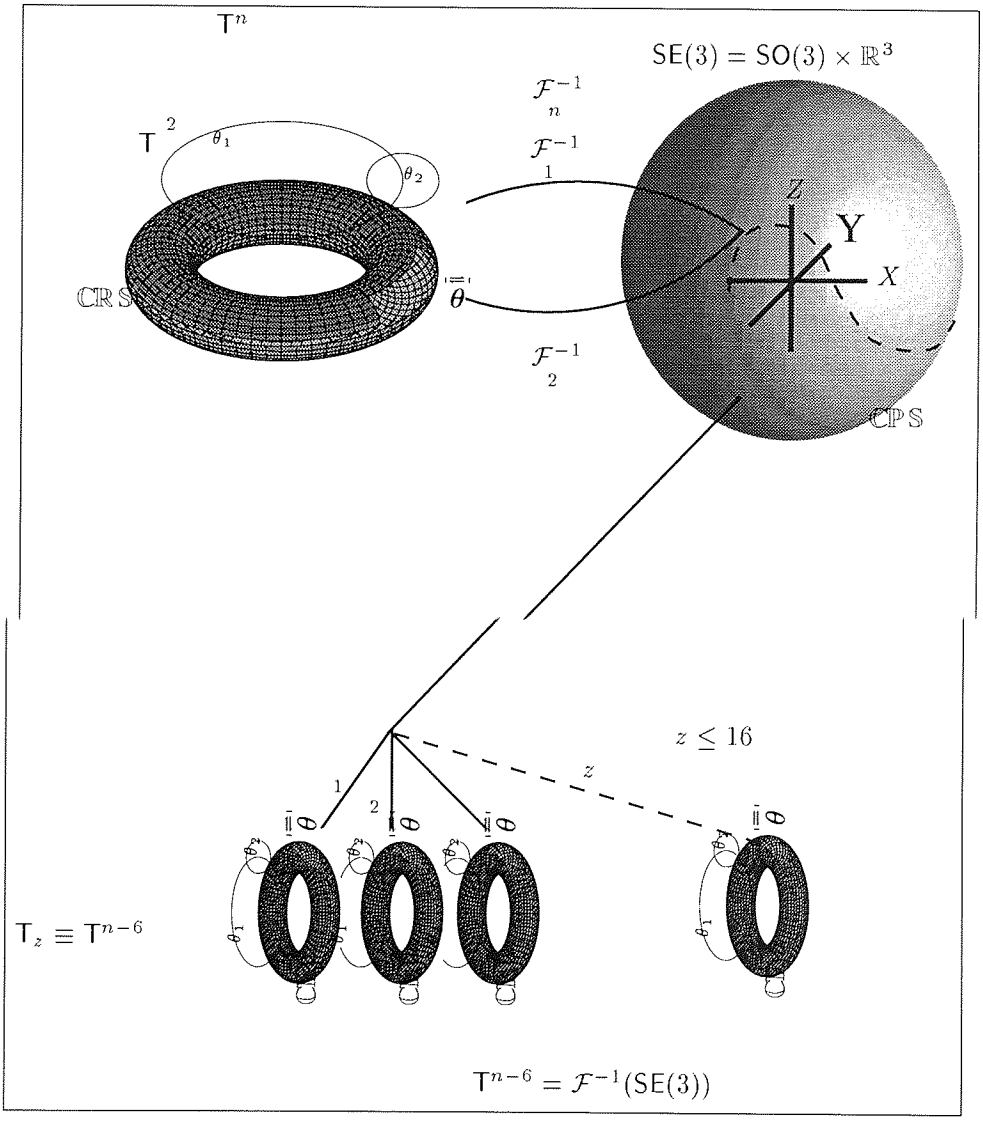

Figure 2.9: The Kinematic Solution Structure of a Dextrous Manipulator with 7 *dof*. The top right sphere and three dimensional cross is a graphical representation of a coordinatisation of SE(3).

$\mathbf{J}^+$ is still full-rank. A particular disadvantage of optimisation methods is that they do not guarantee singularity avoidance irrespective of cost function M_t (Nakamura, 1991). A particular disadvantage of Lyapunov stabilisation is that while it permits additional side constraints to be satisfied, it does not guarantee the primary task of end-effector control with zero error while satisfying secondary tasks (Asada and Slotine, 1986).

Direct inverse functions have few disadvantages. Suitably constructed direct inverse functions may be cyclic in the solution space unlike the previous three methods. If direct inverse solutions can not be derived, the suitability of the solution method depends on which of the particular disadvantages of each solution method is least critical in a given application domain.

Chapter 3

Sequential Calibration of Camera and Manipulator Parameters

If closed form or differential inverse kinematic solutions have been derived for a class of manipulators, solution accuracy depends on measurement of DH link $\mathbf{a}, \boldsymbol{\alpha}, \mathbf{d}$ and joint parameters θ_i. The DH coefficients of $\mathcal{F}^{-1}$ can be re-calibrated if the position and orientation of a manipulator can be measured. Parametric model-based DFA style robotics typically requires inverse kinematic accuracies of the order of $0.2mm$ to $1mm$. This level of inverse kinematic accuracy is generally achieved in two steps:

1. high precision production and measurement of parts and usage of correction parts during each fabrication step.

2. detailed kinematic error determination after fabrication.

In the second case, because of strong kinematic nonlinearities, an explicit correction in analytical form based on errors may not be possible. If this is the case error corrections are stored in software tables. Uniform high precision correction is not usually possible due to the limited number of table values and nonlinear interpolation errors with fixed table quantisation. Iterative correction of manufacturing inaccuracy and determination of software tables are both difficult and account for a large amount of the fabrication costs of manipulators.

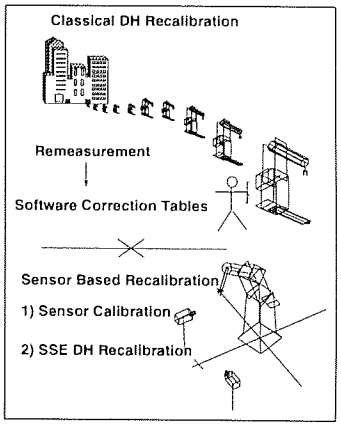

Figure 3.1: Explicit and Sensor-Based Recalibration of Denavit Hartenberg Parameters.

During operation, manipulator geometry deviates over time from the manufactured nominal closed-form solution $\mathfrak{m}_{m1} \rightarrow \mathfrak{m}_{m2}$. DH parameters can change for example as a result of environmental collisions. The standard method in factories to periodically re-calibrate kinematics involves expensive and laborious off-line measurement of DH parameters or re-correction of error correction tables (Stone, 1987). Large kinematic changes cannot be corrected and require defective parts to be repaired.

Explicit parameter re-measurement and part repair are infeasible in remote and/or hazardous environments[1] where:

[1] For instance Mars or inter-planetary (where the operational temperatures on the same manipulator may range from several hundred degrees Fahrenheit facing the sun to close to absolute zero in the shade) or most stringent of all within a nuclear reactor during meltdown where in addition to high temperatures there is much electrical interference.

1. manipulator geometry may change as a result of collision damage,
2. joint encoders may become flawed,
3. joints may freeze in extreme low temperatures,
4. joints may jam in extreme high temperatures.

For these reasons, remote hazardous environments require sensor-based approaches to re-estimation of DH parameters. The goal of research into sensor calibration of models, see (Stone, 1987; Mooring et al., 1991), is to optimise the tradeoff between sensor cost, accuracy, reliability, potential sensing holes, potential sensor damage, and extrinsic sensors as obstacles. This Chapter analyses background research on calibration of forward kinematic equations before deriving an efficient evaluation of sequential:

1. calibration of extrinsic parametric stereo-camera to $\mathfrak{C}$,
2. re-calibration of DH parameters by camera measurements of forward kinematics.

Efficiency of numerical methods for re-calibration of parametric models is important since parametric models of forward kinematics are highly nonlinear multi-dimensional functions. Determination of the DH parameters directly from a DH model is difficult due to the nonlinearity of the relation of end-effector pose to DH parameters. It is computationally expensive to symbolically evaluate parameter calibration error correction gradients based on explicit symbolic evaluation of the product of homogeneous matrices required to compute forward kinematic pose position as a function of DH parameters. To overcome this problem, an error gradient evaluation based on the product rule for derivatives of matrix products is proposed in this Chapter.

In order to recalibrate manipulator parameters of some kinematic convention, the position and orientation of a manipulator in some $\mathfrak{C}$ coordinatisation of SE(3) can be measured by

❶ **contact** tools and devices at defined positions in $\mathfrak{W}$. Special contact tools, e.g. coordinate measuring machines, straight line motion constraint contact equipment (Renders et al., 1991; Everett and Ives, 1993), are obstacles within $\mathfrak{W}$. Only a few high accuracy measurements are possible. Nonlinearity of $\mathcal{F}^{-1}$ makes interpolation difficult.

❷ **distal** sensors within whole sub-workspaces. Measurement accuracy is primarily a function of the distance of manipulator features from the sensors.

(a) $1, 2...m$ sensors **extrinsic** to a manipulator. Extrinsic sensors give a global overview of $\mathfrak{W}$ and the manipulator. They are useful for combining measurement of a manipulator and task constraints such as trajectory collision detection. Kinematic precision based on m extrinsic sensors is bounded above by the minimum distance of a manipulator from the sensors. See (Basu, 1993) for an example of tracking extrinsic camera based kinematic recalibration and (Prenninger et al., 1993; Heeren and Veldpaus, 1992; Newman and Osborn, 1993) for examples of tracking laser interferometry based recalibration.

(b) $1, 2...n$ **intrinsic** sensors. Intrinsic or manipulator-mounted sensors can determine the relative position of objects in $\mathfrak{W}$ with respect to a manipulator. Accuracy of intrinsic sensor measurement increases as a manipulator approaches feature sets.

(c) n intrinsic and m extrinsic sensors.

This Chapter examines the case of extrinsic stereo-camera calibration.

3.1 Calibration of Interior and Exterior Camera Models

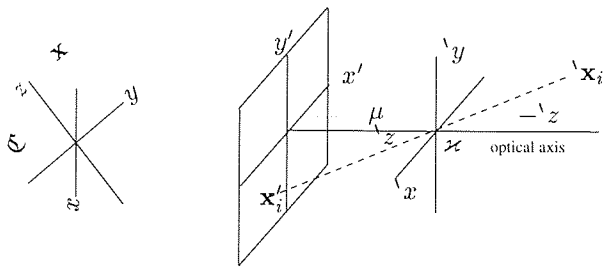

Figure 3.2: The Relation of Image, Camera and Cartesian Coordinate Systems. Light goes from right to left from $\backslash\mathbf{x}_i$ to $\mathbf{x}'_i$ through $\varkappa$.

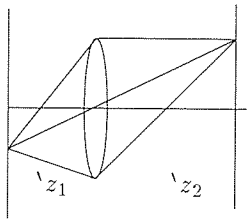

Figure 3.3: The Geometry of a Focal Length Equation.

In Figure 3.2, for a pinhole camera model with pinhole at $\varkappa$, following (Horn, 1986) let:

3.1 Definition (Cartesian, Camera, and Image Coordinate Systems)

1. $\{x, y, z\} \equiv \mathbf{x}_i \in \mathbb{R}^3$ be the position of a point feature in $\mathfrak{C}$.
2. $\{\backslash x, \backslash y, \backslash z\} \equiv \backslash\mathbf{x}_i \in \mathbb{R}^3$ be the position of the same point feature in a coordinate system whose origin is at $\varkappa$. Axes of $\backslash x, \backslash y$ coordinates are parallel to a rectangular camera Charge Coupled Device (CCD) plane and aligned with its edges. μ is the distance form $\varkappa$ to the CCD along the optical axis. The optical axis lies along the axis of the z coordinate.
3. $\{x', y'\} \equiv \mathbf{x}'_i \in \mathbb{R}^2$ be the image coordinates of a projection of $\backslash\mathbf{x}_i$ onto a camera CCD. x', y' are aligned with $\backslash x, \backslash y$.

An ideal lens in Figure 3.3 brings to focus light from points at a distance $-\backslash z_2$ given by (Horn, 1986)

$$\frac{1}{\backslash z_1} + \frac{1}{-\backslash z_2} = \frac{1}{f}, \tag{3.1}$$

where f is the focal length of the lens. If the assumption is made that a camera is in focus for distant objects, $\mu \approx f$. $\mathbf{P}$, a pin-hole perspective transformation, maps a point feature $\backslash\mathbf{x}_i \rightarrow \mathbf{x}'_i$

$$\mathbf{P} : x' = \frac{\backslash x f}{\backslash z} \implies \backslash x = \frac{\backslash z x'}{f}, \tag{3.2}$$

$$\mathbf{P} : y' = \frac{\backslash y f}{\backslash z} \implies \backslash y = \frac{\backslash z y'}{f}. \tag{3.3}$$

A typical model $\mathbf{L}$ of wide-angle radial lens distortion is the polynomial (Horn, 1986)

$$\mathbf{L} : \Delta r' = k_1(r') + k_2(r')^2 + k_3(r')^3 + \ldots, \tag{3.4}$$

where $r' = ||\mathbf{x}'_i||_2$ is the distance of a point $\mathbf{x}'_i$ in the image from the place where the optical axis pierces the image plane and $k_1, k_2, \ldots, k_n$ are lens parameters. Let $\mathcal{M}_1 = \mathbf{LP}$ designate an interior camera model.

An exterior model $\mathcal{M}_2$ maps $\mathbf{x}_i$ to $\backslash\mathbf{x}_i$. Such a general mapping requires 3 parameters to specify positional offset and 3 parameters to specify rotational offset. A general exterior camera model is a homogeneous, for instance DH, transform $\backslash\mathbf{x}_i = \mathbf{T}\mathbf{x}_i$. If a DH orthogonality constraint, i.e. the rotational constraint, on $\mathbf{T}$ is relaxed, an affine transform $\mathbf{A}$ with 15 parameters can model (Horn, 1986):

1. translation error due to inaccurate measurement of displacement of origins between $\mathbf{x}_i, \backslash\mathbf{x}_i$ coordinate systems,

2. rotation due to inaccurate measurement of CCD rotation with respect $`x_i, `y_i$,

3. skewing error due to departure of orthogonality of the CCD,

4. shearing error, due to unequal scaling of image axes $`x_i, `y_i$.

In §5.5, $\mathbf{A}$ is used to implement an affine stereo approximation to the full perspective stereo projection transformation.

Classical calibration (Fu et al., 1987; Horn, 1986) of interior $\mathcal{M}_1$ and exterior $\mathcal{M}_2$ models can be

1. **Absolute** based on explicit measurement of parameters of $\mathcal{M}_1$ and $\mathcal{M}_2$.

2. **Relative** to samples $\mathbf{x}_i, \mathbf{x}'_i, i = 1 \ldots n$ gathered with n manipulator movements where $\mathbf{x}_i = \mathcal{F}(\theta_i), \mathbf{x}'_i = \Psi(\mathcal{F}(\theta_i))$, where Ψ is a sensor function, for instance stereo-camera. A sensor function is defined on a Cartesian space by specification of sets of calibration features and definition of the function graph from the span of these features of Cartesian space to measurements in the image coordinate system.

   (a) if $\mathcal{M}_1$ does not contain $\mathbf{L}$ (which is equivalent to zero lens distortion $\mathbf{L} = \mathbf{I}$) or $\mathbf{L}$ parameters $k_1, \ldots, k_n$ are known for transformation $\mathbf{L} : \mathbf{x}'_a \rightarrow \mathbf{x}'_b$, $\mathbf{L}$ may be inverted $\mathbf{L}^{-1}$. In this case, the remainder of $\mathcal{M}_1 \equiv \mathbf{PA}$ can be calibrated by pseudoinversion over vector samples $\mathbf{x}_i$ organised into matrix form. Row i of matrix $\mathbf{X}$ is equal to $\mathbf{x}_i$, row i of matrix $\mathbf{X}'$ is equal to $\mathbf{x}'_i$ where $\mathbf{X}' = \mathbf{LPAX}$, and $\mathbf{PA} = \mathbf{L}^{-1}\mathbf{X}'\ \mathbf{X}^{+}$.

   (b) If $\mathbf{L}$ parameters are unknown, lens nonlinearity implies that parameters of $\mathbf{L}$ and $\mathbf{PA}$ must be determined by gradient-based iterative minimisation of an error function

$$E = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{LPA}\mathbf{x}_i - \mathbf{x}'_i)^2. \tag{3.5}$$

3.2 Calibration of Stereo-Camera Models

After calibration of interior and exterior camera models, in order to use camera measurements of feature sets to re-calibrate closed-form solvable kinematics, $\mathcal{M}_1, \mathcal{M}_2$ must be inverted to map measurements $\mathbf{x}'_i$ of a manipulator feature location to $`\mathbf{x}_i$ and then to $\mathbf{x}_i$ (Fu et al., 1987; Horn, 1986; Vernon, 1991). $\mathcal{M}_2^{-1} = \mathbf{A}^{-1}$ but $\mathcal{M}_1^{-1}$ is not determined since $\mathbf{P} : \mathbb{R}^3 \rightarrow \mathbb{R}^2 : `\mathbf{x}_i \rightarrow \mathbf{x}'_i$ is many-to-one and thus not invertible. A single image, $\mathbf{x}'_i$, does not provide enough information to reconstruct $`\mathbf{x}_i$ coordinates (Fu et al., 1987). $`\mathbf{x}_i$ may computed from $\mathbf{x}'_i$ using (Horn, 1986; Fu et al., 1987)

1. a feature **model**. This implies a priori knowledge of n point features $`\mathbf{x}^1, `\mathbf{x}^2, \ldots `\mathbf{x}^n$. Usually used in practice instead of point features are (Horn, 1986)

   (a) **local** descriptors such as object edges or corners.

   (b) **global** descriptors such as geometric area and moments, or optic flow field.

   Given a feature model, for instance a uniform spherical object as a feature with area $\mathfrak{I}'$ as well as centroid x', y',

$$`z = f\sqrt{\frac{`\mathfrak{I}}{\mathfrak{I}'}} - f, \tag{3.6}$$

   where $`\mathfrak{I}$ is defined as the projected area of an object when placed at f.

2. multiple image

   (a) **parallax** between n manipulator movements and ${}^{\backprime}\mathbf{x}_i$ observed by an intrinsic camera as $\mathbf{x}'_1, \mathbf{x}'_2, \ldots \mathbf{x}'_n$.

   (b) **stereo-camera** triangulation of **corresponding** features ${}_1\mathbf{x}'_i, {}_2\mathbf{x}'_i$ for camera one and two.

Calibration of stereo-cameras is a classic photogrammetric problem. See (Duda and Hart, 1973). If stereo camera image coordinate systems are aligned and co-planar with centres separated by a baseline distance b, $\mathbf{Z}$ is a mapping to compute ${}^{\backprime}z$ by triangulation if corresponding features ${}_1\mathbf{x}'_i, {}_2\mathbf{x}'_i$ can be matched (Fu et al., 1987)

$$\mathbf{Z} : {}^{\backprime}z_i = \frac{fb}{{}_1x'_i - {}_2x'_i}. \tag{3.7}$$

Substitution of (3.6) or (3.7) in (3.2),(3.3) uniquely specifies ${}^{\backprime}\mathbf{x}_i$. To extend a coplanar stereo-camera model to the non-coplanar stereo-camera model $\mathbf{S}$ requires a simultaneous solution to ${}_1\mathcal{M}_1^{-1}, {}_2\mathcal{M}_1^{-1}$. An affine stereo approximation of $\mathbf{S}$ derived in this section is also denoted $\mathbf{S}$ to avoid superscipt and subscript clutter. If $\mathbf{x}_i$ is represented in homogeneous coordinates

$$ {}_1\mathbf{L}^{-1}\ {}_1\mathbf{x}'_i = {}_1\mathbf{P}\ {}_1\mathbf{A}\mathbf{x}_i, \tag{3.8}$$

$$ {}_2\mathbf{L}^{-1}\ {}_2\mathbf{x}'_i = {}_2\mathbf{P}\ {}_2\mathbf{A}\mathbf{x}_i. \tag{3.9}$$

If $\mathbf{x}_i$ is row i of $\mathbf{X}$ and ${}^*\mathbf{x}_i$ is column i of ${}^*\mathbf{X}$, where

$${}^*\mathbf{x}_i = [{}_1\mathbf{L}^{-1}\ {}_1\mathbf{x}'_i, {}_2\mathbf{L}^{-1}\ {}_2\mathbf{x}'_i]^T, \tag{3.10}$$

then

$$\mathbf{X} = \mathbf{S}\ {}^*\mathbf{X}, \tag{3.11}$$

$$\mathbf{S} = \mathbf{X}\ {}^*\mathbf{X}^{-1}. \tag{3.12}$$

(Horn, 1986) concludes a derivation of the full perspective stereo[2] model with the observation that determination of full perspective $\mathbf{S}$ is straightforward if camera measurements of features can be matched. Robust matching of corresponding stereo features of a manipulator in a general scene is the principal obstacle to stereo-image based manipulator kinematic recalibration. Computation of stereo feature correspondence is complicated by possible feature occlusion by a manipulator and vice versa.

In the absence of easy-to-segment features on a manipulator such as LEDs for local feature matching, global stereo-features ${}_1\mathbf{x}'_i$, ${}_2\mathbf{x}'_i$ such as edges or intensity patches can be matched by correlation and energy function minimisation methods (Horn, 1986). These methods for global feature matching are computationally intense and comparatively not as robust as local easy-to-segment feature matching methods except in the case of local feature occlusion.

# cameras	#features
1	4
2	3
3	2
4	1

Table 3.1: The Minimal Number of Features for Measurement of Position and Orientation.

Once stereo-image measurements of a point feature on a manipulator have been calibrated to $\mathfrak{C}$, the positional location $\mathbf{x}_i$ of the end effector can be determined by

$$\mathbf{x}_i = \mathbf{S}\ [{}_1\mathbf{x}'_i, {}_2\mathbf{x}'_i]^T. \tag{3.13}$$

[2] $\mathbf{S}$ is useful for scene analysis if three dimensional information of objects can be recovered from images and mapped to $\mathfrak{C}$ for parametric, for instance deformable, model matching. $\mathfrak{C}$ models can be mapped via an inverse kinematic function into a free space (see (Latombe, 1991)) representation for path planning.

Table 3.1 gives the number of non-collinear point features required by sets of cameras to uniquely specify the position and orientation of a manipulator in $\mathfrak{C}$ (see (Horn, 1986; Craig, 1989; Fu et al., 1987; Samson et al., 1991)). Assuming a stereo-camera system $\mathbf{S}$ has been calibrated using one feature $\mathbf{x}_i^1$ of manipulator position, if two other features $\mathbf{x}_i^2, \mathbf{x}_i^3$ are identified they are sufficient to determine position and orientation of the manipulator in $\mathfrak{C}$ by

$$\mathbf{x}_i^{1,2,3} = \{\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{x}_i^3\} = \bigcup_{j=1}^{3} \mathbf{S}\ [{}_1\mathbf{x}'^j_i, {}_2\mathbf{x}'^j_i]^T. \tag{3.14}$$

$\mathbf{x}^{1,2,3}$ specify the position and orientation of a manipulator in $\mathfrak{C}$, i.e. is an element of a representation of SE(3). In the next section, $\mathbf{x}_i$ is redefined to be the position and orientation of a manipulator as specified by $\mathbf{x}_i^{1,2,3}$.

3.3 Optimal Re-calibration of DH Parameters with Products of DH Matrix Derivatives

After stereo-cameras have been calibrated to $\mathfrak{C}$ by one of the above methods, during subsequent operation, $\mathcal{M}_1, \mathcal{M}_2$ or F^{-1} may change. To determine which which of $\mathcal{M}_1, \mathcal{M}_2$ or F^{-1} has changed:

1. If $\mathcal{F}^{-1}$ continues to function well during technological operations, it may be assumed that the DH parameters are correct $\Longrightarrow$ $\mathcal{M}_2, \mathcal{M}_1$ have changed.

2. If an absolute reference feature $\mathbf{x}_a$ is placed in the environment observable by the cameras, this can be checked to see if $\mathcal{M}_2, \mathcal{M}_1$ have changed. $\mathbf{x}_a$ can be used to re-calibrate $\mathcal{M}_2, \mathcal{M}_1$. Without $\mathbf{x}_a$, if cameras and manipulator are simultaneously de-calibrated, cameras can be re-calibrated using (3.5) with respect to the new DH parameters. In §4, a method is developed which avoids the necessity for $\mathbf{x}_a$, and can accomplish absolute non-explicit camera and DH calibration simultaneously assuming that f, the focal length parameter, and some kinematic parameters are correct.

3. If $\mathcal{F}^{-1}$ does not function well during technological operations, it may be assumed that DH parameters are incorrect. Samples of $\mathbf{x}'_i$ and $\mathbf{x}_i$ may be used to re-calibrate DH parameters governing F^{-1}.

From the previous section, stereo-cameras have been calibrated with respect to a given set of DH parameters. If DH parameters $\mathbf{w} \equiv \{\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}\}$ change, the difference between computed forward kinematics $\mathbf{x}^i \equiv \mathbf{T}|\boldsymbol{\theta}_i$, where $\mathbf{T}$ is homogeneous transform notation defined in §2.1 evaluated at $\boldsymbol{\theta}_i$ and calibrated stereo-camera measurements $\mathbf{x}_i$ of a manipulator can be used to re-calibrate $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$.

Let $\mathbf{x}_i$ be a point in a m dimensional Cartesian space. $\mathbf{x}_i$ may be represented in various coordinate systems. One convenient coordinate system is as a vector $\mathbf{x}_i$. For instance in the previous section, Cartesian space was a three dimensional space and $\mathbf{x}_i$ was defined to be a three dimensional vector defined with respect to a right handed orthonormal coordinate system. In this section Cartesian space is the Special Euclidean group. One coordinate system for Cartesian space is the homogeneous coordinate system and another is a coordinate system based for instance on Fixed Euler angles and position vector. These representations are locally but not globally homotopic. A location of a manipulator in Cartesian space $\mathbf{x}_i$ can be defined as a vector or as a homogeneous transform if one wishes to exploit matrix algebra for representation of functions. For this reason, in the following equations, $\mathbf{x}_i$ denotes

a Cartesian target position where it is understood that this can be represented by a homogeneous transform $\mathbf{T}_i$.

A 6 *dof* spatial manipulator has 18 $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$ parameters. To calibrate 18 DH parameters, at least $z = 18/6$ non-collinear $\mathbf{x}_i$ samples are required. E, a quadratic error function is defined for $\{\mathbf{x}_i, \mathbf{x}^i = F(\boldsymbol{\theta}_i)\}$, $i = 1, \ldots, z$

$$E = \frac{1}{2}\sum_{i=1}^{z}(F(\boldsymbol{\theta}_i) - \mathbf{x}_i)^2. \tag{3.15}$$

where $\mathbf{x}^i = F(\boldsymbol{\theta}_i)$ is the forward kinematic homogeneous transform product. If $\mathbf{x}_i$ is represented as a homogeneous transform $\mathbf{T}_i$, for each element of the first three rows, $m = 1, \ldots, 12$, of homogeneous transform T_i

$$E = \frac{1}{2}\sum_{i=1}^{z}\sum_{m=1}^{12}(x^m - x_i^m)^2. \tag{3.16}$$

(3.16) is re-expressed in homogeneous transform notation as

$$E = \sum_{i=1}^{z}\sum_{\epsilon\in\mathbf{T}}(\mathbf{T}^i - \mathbf{T}_i)^{2*}. \tag{3.17}$$

where $2*$ denotes element wise matrix squaring. For an n *dof* manipulator assuming that no DH parameters are correct, symbolic expansion of the first three rows of $\mathbf{T} = \prod_{j=1}^{n}\mathbf{T}_j$ produces a set of 12 equations in 18 unknowns. The 12 equations are highly nonlinear and contain hundreds of terms and

> "determining kinematic parameters directly from kinematic model given measurement data, although appealing both theoretically and practically, is difficult since pose components of robot end-effector are in general nonlinear function of the robot link parameters." (Zhang and Roth, 1993). pg. 174.

On a Sun SPARCstation 10, Matlab calls to the Maple symbolic kernel can only evaluate $\prod_{j=1}^{4}\mathbf{T}_j$. $\mathbf{T}_j$ equations could be derived by transform-wise composition of sub-products for each output element. However even if an analytical expression of $\prod_{j=1}^{6}\mathbf{T}_j$ is derived and evaluated for 3 non-collinear $\boldsymbol{\theta}_i$, it would be computationally prohibitive to

1. simultaneously solve the resulting 36 equations in 18 unknowns,
2. minimise by least squares the 36 or reduced 18 equations based on explicit function and gradient formula evaluation

in order to determine the values of $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$. Instead, it is more tractable to use numeric transform evaluation of $\prod_{j=1}^{6}\mathbf{T}_j$ and error gradients for each sample $\boldsymbol{\theta}_i$ in a nonlinear least squares minimisation procedure. Let $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$ be re-expressed in general parametric notation as $w_{jk} \in \mathbf{w}$ such that j marks the transform $\mathbf{T}_j$ in which $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$ appears and $\forall j, w_{j1} = \alpha_j, w_{j2} = d_j, w_{j3} = a_j$. From §2.3, for each sample $\mathbf{x}_i \in \mathfrak{C}$

$$\mathbf{x}^i = f(\boldsymbol{\theta}_i, \boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}) = \prod_{j=1}^{n}\mathbf{T}_j. \tag{3.18}$$

$\mathbf{T}_j$ is a function of w_{jk} and θ_j. Let $\boldsymbol{\theta}_i, i = 1, \ldots, z$ generate z manipulator movement samples and let $\mathbf{x}_i \equiv \mathbf{T}_i$ be corresponding sensor measurements of manipulator position mapped to $\mathfrak{C}$ by the methods

of the previous section. For a given i let $\mathbf{x}^i = \mathbf{T}^i = \prod_{j=1}^{n} \mathbf{T}_j \equiv f(\boldsymbol{\theta}_i; \mathbf{w})$. Once θ_j for sample i is evaluated within $\mathbf{T}_j$, $\mathbf{x}^i \equiv \prod_{j=1}^{n} f_j(\mathbf{w}_j)$, where f_j is defined in the (Fu et al., 1987) DH convention.

$$f_j \equiv \mathbf{T}_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \cos\alpha_j & \sin\theta_j \sin\alpha_j & a_j \cos\theta_j \\ \sin\theta_j & \cos\theta_j \cos\alpha_j & -\cos\theta_j \sin\alpha_j & a_j \sin\theta_j \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.19}$$

The error gradient for parameter w_{jk} is defined by

$$\frac{\partial E}{\partial w_{jk}} = \sum_{i=1}^{z} \sum_{\epsilon \in \mathbf{T}} (\mathbf{T}^i - \mathbf{T}_i) \frac{\partial \mathbf{T}^i}{\partial w_{jk}}. \tag{3.20}$$

Transform derivatives by the derivative product rule are

$$\frac{\partial \mathbf{T}^i}{\partial w_{jk}} = \mathbf{T}_1 \mathbf{T}_2 \ldots \frac{\partial \mathbf{T}_j}{\partial w_{jk}} \ldots \mathbf{T}_n. \tag{3.21}$$

Each transform $\mathbf{T}_j$ has a partial gradient transform with respect to α_j, d_j, a_j

$$\frac{\partial \mathbf{T}_j}{\partial \alpha_j} = \begin{bmatrix} 0 & \sin\theta_j \sin\alpha_j & \sin\theta_j \cos\alpha_j & 0 \\ 0 & -\cos\theta_j \sin\alpha_j & -\cos\theta_j \cos\alpha_j & 0 \\ 0 & \cos\alpha_j & -\sin\alpha_j & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.22}$$

$$\frac{\partial \mathbf{T}_j}{\partial d_j} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.23}$$

$$\frac{\partial \mathbf{T}_j}{\partial a_j} = \begin{bmatrix} 0 & 0 & 0 & \cos\theta_j \\ 0 & 0 & 0 & \sin\theta_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.24}$$

If d_j is variable, i.e. for a prismatic joint,

$$\frac{\partial \mathbf{T}_1}{\partial \theta_j} = \begin{bmatrix} \sin\theta_1 & -\cos\theta_1 \cos\alpha_1 & \cos\theta_1 \sin\alpha_1 & -a_1 \sin\theta_1 \\ \cos\theta_1 & -\sin\theta_1 \cos\alpha_1 & \sin\theta_1 \sin\alpha_1 & a_1 \cos\theta_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.25}$$

The error Hessian is defined as

$$\frac{\partial^2 E}{\partial w_{jk} \partial w_{j'k'}} = \mathbf{H}_{\{jk\}\{j'k'\}} = \sum_{i=1}^{z} \sum_{m=1}^{12} \left[\frac{\partial x^m}{\partial w_{jk}} \frac{\partial x^m}{\partial w_{j'k'}} + (x_i^m - x^m) \frac{\partial^2 x^m}{\partial w_{jk} \partial w_{j'k'}} \right], \tag{3.26}$$

expressed in transform notation as

$$= \sum_{i=1}^{z} \sum_{\epsilon \in \mathbf{T}} \left[\frac{\partial \mathbf{T}^i}{\partial w_{jk}} .* \frac{\partial \mathbf{T}^i}{\partial w_{j'k'}} + (\mathbf{T}^i - \mathbf{T}_i) .* \frac{\partial^2 \mathbf{T}^i}{\partial w_{jk} \partial w_{j'k'}} \right], \tag{3.27}$$

where by a second application of the product rule for transform derivatives

$$\frac{\partial^2 \mathbf{T}^i}{\partial w_{jk} \partial w_{j'k'}} = \mathbf{T}_1 \mathbf{T}_2 \ldots \frac{\partial \mathbf{T}_{j'}}{\partial w_{j'k'}} \ldots \frac{\partial \mathbf{T}_j}{\partial w_{jk}} \ldots \mathbf{T}_n. \tag{3.28}$$

Descent in the Newton direction

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{H}^{-1} [\frac{\partial E}{\partial w_1} \ldots \frac{\partial E}{\partial w_{3n}}]^T \tag{3.29}$$

minimises E, i.e. calibrates the DH parameters. Re-calibration of DH parameters for a 6 *dof* spatial manipulator requires for each sample $\boldsymbol{\theta}_i$ the sub-evaluation of 18 partial gradient transforms $\frac{\partial \mathbf{T}_j}{\partial w_{jk}}$ and 324 partial Hessian transforms $\frac{\partial^2 \mathbf{T}^i}{\partial w_{jk} \partial w_{j'k'}}$ each the result of 6 products of 4×4 homogeneous transforms. Since evaluation of $\frac{\partial^2 \mathbf{T}^i}{\partial w_{jk} \partial w_{j'k'}}$ accounts for 80% of computation requiring 497664 floating point additions and multiplications per calibration sample, the following numerical optimisations further reduce complexity of error Hessian evaluation:

1. tabulation of (3.28) sub-products reduces Hessian evaluation costs 4-fold.
2. use of symmetry - $\frac{\partial^2 \mathbf{T}^i}{\partial w_{jk} \partial w_{j'k'}} = (\frac{\partial^2 \mathbf{T}^2}{\partial w_{kj} \partial w_{k'j'}})^T$ reduces Hessian evaluation costs two fold.
3. tabulation of common trigonometric evaluations reduces Hessian evaluation costs as a function of the costs of the algorithm for trigonometric function evaluation.

Partial Hessian evaluation can be further minimised by taking into account transform and transform derivative sparsity.

A redundant manipulator introduces extra DH parameters, into transform $\mathbf{T}_r$, for calibration. Once cameras are calibrated, a redundant arm can be re-calibrated by

1. Selection of a joint θ_r to parameterise redundancy, for instance a redundant second elbow.
2. Selection of a point θ_r^* in θ_r range (for $\mathbf{R}$ joints, $\theta_r^* \in (0 - 2\pi]$ if there are no joint limits).
3. Once θ_r^* is fixed, $\mathbf{T}_r$ is a constant matrix and samples $\mathbf{x}_i, \mathbf{x}_i'$ do not depend on θ_r. DH parameters for other $\mathbf{T}_j$, $j \neq r$ can be calibrated by the method just described for $\prod_{j=1}^{n} \mathbf{T}_j$.
4. When DH parameters have been calibrated for $\mathbf{T}_j, j \neq r$, joints θ_j, $j \neq r$ can be fixed at appropriate values within their ranges. This turns $\mathbf{T}_j, j \neq r$ into constant matrices. If θ_r is now released and used to generate samples $\mathbf{x}_i, \mathbf{x}_i'$, these do not depend on θ_j, $j \neq r$. DH parameters of $\mathbf{T}_r$ can now be calibrated by the method described above.

3.3.1 A Numerical Experimental Validation of the Re-calibration Algorithm

(3.15)-(3.29) were implemented with a simulated 6 *dof* PUMA 560 model for numerical experiments to re-calibrate $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{a}$ parameters. Implementation of (3.17), (3.27) was verified by central difference[3] approximations

$$\frac{\partial E}{\partial w_i} \approx \frac{E(w_i + \Delta) - E(w_i - \Delta)}{2\Delta}, \tag{3.30}$$

$$\frac{\partial^2 E}{\partial w_i \partial w_j} \approx \frac{\frac{dE(w_i + \Delta)}{dw_j} - \frac{dE(w_i - \Delta)}{dw_j}}{2\Delta}. \tag{3.31}$$

[3]These could also be used to compute the gradients but they are $O(\dim(\mathbf{w})^3)$ complexity to evaluate.

$\mathbf{H}^{-1}$ in (3.29) was inverted by an implementation of Gauss Jordan matrix inversion ($O(\dim(\mathbf{w})^3)$). (3.29) was minimised by cubic spline interpolation line search, see (Press et al., 1992), pg. 384-386. Approximation samples must span SE(3) for re-calibration of DH parameters governing both positioning and orientation. Since $\mathcal{F}^{-1}$ is initially incorrect, a spanning set $\mathbf{x}_i$ of $\mathfrak{C}$ can be generated by redundant sampling to create an over-determined minimisation problem. The performance of the

link	joint	twist	length	offset
i	θ_i	α_i	a_i(m)	d_i (m)
1	θ_1	1.5774	0.0156	0.0279
2	θ_2	0.0014	0.4567	0.0254
3	θ_3	−1.5504	0.0213	0.1412
4	θ_4	1.5912	0.0016	0.4346
5	θ_5	−1.5428	0.0159	0.0196
6	θ_6	0.0115	0.0201	0.0125

Figure 3.4: A De-calibrated PUMA 560 DH Table.

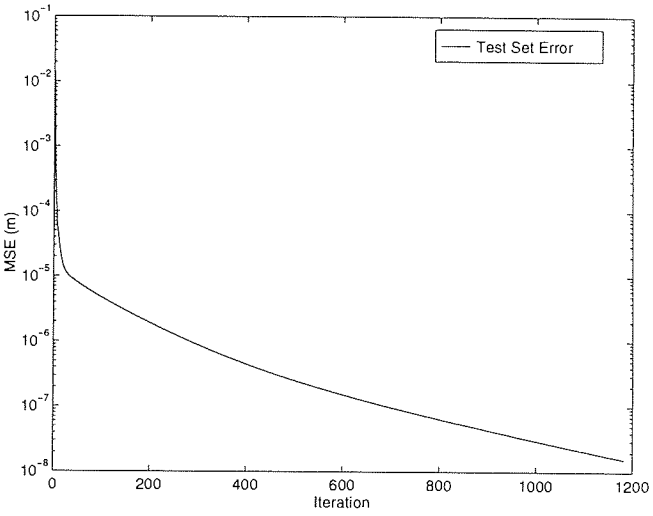

Figure 3.5: Test Error for Re-calibration of the PUMA 560 DH Parameters.

numerical DH parameter recalibration algorithm was tested on a set of correct and incorrect DH models over a set of 30 experiments. No optimal numerical experimental design was developed so only the results of a representative numerical experiment to re-calibrate $\mathcal{F}_1$, the PUMA 560 of Table 3.4, to $\mathcal{F}_2$, the PUMA 560 of Table 2.1, are presented in Figure 3.5. For this experiment, 64 calibration samples were generated in a range $j = 1, \ldots 6 : \forall \theta_j, \theta_j = \{0, \pi/10\}$. 2 samples per joint permuted by six joints $= 2^6$ samples of $\boldsymbol{\theta}_i$. Calibration data is generated by $\mathcal{F}_2(\boldsymbol{\theta}_i)$. The parameters of $\mathcal{F}_1$ were decalibrated from those of $\mathcal{F}_1$ by a uniform random $\pm 3cm$ de-calibration of each DH parameter. Such de-calibration is within the locus of convergence of (3.29) to the global minimum. No noise was added to the samples but real re-calibration could be done with this order of samples to average over sensor noise and to ensure non-collinear spread of calibration data. The numerical recalibration algorithmic convergence criteria were that:

1. $MSE < 10^{-6}m^2$,
2. and $\frac{dE}{dw} < 10^{-7}m^2, \forall w$.

From Figure 3.5 shows that the PUMA 560 inverse kinematic function is very nonlinear. Approximately 250 calibration iterations were needed to bring MSE precision to $10^{-6}m^2$. Figure 3.5 shows an initial step descent of MSE to $10^{-5}m^2$ followed by a slow convergence towards the limits of numerical round-off. From Figure 3.5, real re-calibration of MSE to measurable quantisation limits of images of $10^{-2}m^2 \rightarrow 10^{-3}m^2$ would require few iterations. Analysis of the quantisation limits of sensor measurements provides an estimate of the MSE at the global minimum of model calibration error. See §6.1.2.1.

To progress from testing of algorithmic convergence for recalibration of a few points in DH parameter space to determination of average measures of rates of convergence for different classes of

manipulators for those DH sub-manifolds that result in calibration convergence to a global error minimum for given sets of calibration data requires a large increase in numerical experimental design complexity. The other 29 numerical experiments indicate that in the case of real de-calibration, of the order of $\max \pm 4cm$, models initialised with nominal DH parameters are unlikely to be beyond the concave well of the global minimum corresponding to correct DH parameters. In order to prevent non-positive Hessian eigenvalues in (3.29), in the theoretical case of a DH model initialised far from the global minimum calibration, (3.29) could be supplemented with Levenberg Marquhardt regularisation. If $\mathbf{H}^{-1}$ in (3.29) is replaced with $(\mathbf{H} + \lambda \mathbf{I})^{-1}$, inversion rank is governed by λ. λ methods are known as model trust methods because a model is only trusted in a small region around the search point. Large λ implies small trust. See (Press et al., 1992).

Chapter 4

Simultaneous Calibration of Camera and Manipulator Parameters

The previous chapter derived an efficient method for sequential calibration of stereo-camera and kinematic parametric models. During debugging of numerical experiments it became apparent that it is also possible to simultaneously absolutely calibrate subsets of parameters of both models. Given a manipulator feature[1] $\mathbf{x}_f$ for measurement by extrinsic stereo-cameras, by fixing a subset of parameters in both camera and DH models, the remaining DH parameters governing 3 *dof* positioning, $\mathbf{T} = \prod_{j=1}^{3} \mathbf{T}_j \mathbf{T}_f$ and camera model $\mathbf{A}$ parameters can be simultaneously absolutely calibrated. A simultaneous calibration of a combined kinematic-camera system containing parameters $\mathbf{w}$ does not have a unique calibration solution unless some model parameters $\mathbf{w}_c$ are constrained. If $\mathbf{w}_c$ are not constrained, then a manifold of $\dim(\mathbf{w}_c)$ embedded in $\mathbf{w}$ of calibration solutions is defined. A simultaneous absolute calibration of combined model parameters may have advantages over sequential calibration if certain DH or camera parameters may be less likely to change than others. In this chapter, heuristic numerical experimentation determines parametric subspaces of combined camera-kinematic system which when fixed lead to well defined simultaneous absolute calibration of the remaining parameters of combined models.

4.1 Error Measures for Simultaneous Model Calibration

Let

1. $\mathbf{x}'_i = [{}_1\mathbf{x}'_i \; {}_2\mathbf{x}'_i]$ be feature coordinates in camera one and two corresponding to manipulator location for movement i.

2. $\mathbf{S} : \mathbb{R}^4 \rightarrow \mathbb{R}^3$ be a stereo-camera model that maps a single feature to $\mathfrak{C}$.

[1] A manipulator feature has a DH transform $\mathbf{T}_f$ with fixed θ displacement from a wrist or Tool Centre Point (TCP) origin. $\mathbf{T}_f$ must be known a priori. In simulation it is assumed to be a unit transform for simplification of formulas.

With fixed or un-fixed model parameters, combined DH and camera model parameters may be calibrated by minimisation of an error measure defined in stereo image space

$$E_1 = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{A}(F(\theta_i)) - \mathbf{x}_i')^2 = (\mathbf{A}(F(\theta_i)) - \mathbf{x}_i')^T(\mathbf{A}(F(\theta_i)) - \mathbf{x}_i'), \tag{4.1}$$

or directly in $\mathfrak{C}$

$$E_2 = \frac{1}{2}\sum_{i=1}^{n}(F(\theta_i) - \mathbf{S}\mathbf{x}_i')^2 = (F(\theta_i) - \mathbf{S}\mathbf{x}_i')^T(F(\theta_i) - \mathbf{S}\mathbf{x}_i')^T, \tag{4.2}$$

or indirectly in $\mathfrak{C}$ by a Mahalanobis style measure

$$E_3 = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{S}(\mathbf{A}(F(\theta_i)) - \mathbf{x}_i'))^T(\mathbf{S}(\mathbf{A}(F(\theta_i)) - \mathbf{x}_i')) \tag{4.3}$$

$$= (\mathbf{A}(F(\theta_i)) - \mathbf{x}_i')^T\mathbf{S}^T\mathbf{S}(\mathbf{A}(F(\theta_i)) - \mathbf{x}_i'). \tag{4.4}$$

$F(\theta_i)$ denotes $\mathbf{T} = \prod_{j=1}^{n}\mathbf{T}_j$ evaluated at θ_i and $\mathbf{A}(F(\theta_i))$ denotes $\mathbf{A}_1F, \mathbf{A}_2F$, i.e. the composition of the forward kinematic transform product with the two camera transforms $\mathbf{A}_1, \mathbf{A}_2$. In (4.1), $\mathbf{x}_i'$ measurements contain noise while (4.2) propagates measurement noise to Cartesian space via $\mathbf{S}$ with $\mathbf{S}\mathbf{x}_i'$. E_3 is a measure of the stereo-camera model $\mathbf{S}$ projection to Cartesian space of the error between the image measurements and the projection of the outputs of F, the forward kinematic model, to image space by the camera models $\mathbf{A}$. E_3 involves simultaneous calibration of three models instead of two models of E_1, E_2. More combined parameters can be absolutely calibrated by either E_1 or E_2 than E_3. Hence E_3 is not analysed further.

Minimisation of E_2 in (4.2) can result in loss of rank of $\mathbf{S}$ or F if parameter initialisation is far from the solution. For instance, when Cartesian model $\mathbf{x}_i = \prod_{j=1}^{3}\mathbf{T}_j$ and stereo model $\mathbf{S}$ are both of rank 0, SSE=0. Model loss of rank during minimisation can be avoided by regularisation of E_2 with λE_4 where λ is a regularisation hyper-parameter. At least two regularisers E_4 are possible:

1. inverse Hessian regularisation by a second-order nonlinear least mean squares minimisation procedure, e.g. Levenberg Marquhardt. See § 3.3) and c.f. (Bennet et al., 1991). This is a good straightforward standard solution.

2. It is also possible to derive a special regulariser. Let

$$\sigma_{\mathbf{x}}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(F(\theta_i) - \frac{1}{n}\sum_{i=1}^{n}F(\theta_i))^2 \tag{4.5}$$

be the output variance of F evaluated over n samples of θ_i, and similarly for the variance of the stereo-model

$$\sigma_{\mathbf{S}\mathbf{x}'} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{S}\mathbf{x}_i' - \frac{1}{n}\sum_{i=1}^{n}\mathbf{S}\mathbf{x}_i')^2. \tag{4.6}$$

E_4 is a regulariser

$$E_4 = \frac{\sigma_{\mathbf{x}}^2 - \sigma_{\mathbf{S}\mathbf{x}'}^2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{S}\mathbf{x}'}^2}. \tag{4.7}$$

(4.7) is minimised when the output variances of $\mathcal{F}$ and $\mathbf{S}$ are similar. The denominator term in (4.7) prevents output variance of the two models decaying to zero. To use E_4 as a regulariser it is necessary to compute partial error derivatives $\frac{\partial E_4}{\partial w_k}$ where w_k is a parameter of $\mathcal{F}$ or $\mathbf{S}$. This involves expansion of a quotient derivative and chain derivatives for $\frac{\partial \mathcal{F}}{\partial w_k}$ (see §3.3) and $\frac{\partial \mathbf{S}}{\partial w_k}$. While E_4 is a potentially interesting regulariser, its properties are not explored in this work because minimisation of (4.1) avoids:

1. possible model rank loss, i.e. the need for regularisation with the associated problem of determination of λ.

2. estimation of propagation of pixel quantisation noise.

Hence (4.1) is examined in more detail in the remainder of the Chapter.

4.2 Equi-output Parameter Groups in the Composition of Camera and DH Models

A combined model $\mathbf{A}(\mathcal{F}(\boldsymbol{\theta}_i))$ has groups of equi-output transformation symmetries of its model parameters $\mathbf{w}$. To observe a simple case, consider a combined camera and manipulator model $\prod_{i=1}^{6} \mathbf{T}_i \mathbf{A}$ where $\mathbf{A}$ is an affine stereo projection. If the DH parameters of $\mathbf{T}_6$ are modified by a transformation $\mathbf{T}_*$ such that $\mathbf{T}'_6 = \mathbf{T}_6 \mathbf{T}_*$, then the following systems are output equivalent

$$\prod_{i=1}^{6} \mathbf{T}_i \mathbf{A} = \prod_{i=1}^{5} \mathbf{T}_i \mathbf{T}'_6 \mathbf{A}_* \tag{4.8}$$

where $\mathbf{A}_* = \mathbf{T}_*^{-1} \mathbf{A}_*$. An equi-output transformation g of $\mathbf{w}_a$, $\mathbf{w}_b = g(\mathbf{w}_a)$ leaves the output of a model $\mathbf{A}(\mathcal{F}(\boldsymbol{\theta}_i))$ unchanged. Thus for any set of measurements $(\boldsymbol{\theta}_i, \mathbf{x}'_i)$, there is a group of combined model parameters $(\mathbf{w}_a, \mathbf{w}_b, \ldots, \mathbf{w}_z)$ that minimise (4.1). For simultaneous calibration of both camera and manipulator kinematic parameters, the combined model symmetry group must be null. This occurs if a subset of the combined model parameters $\mathbf{w}_f$ are fixed. In this chapter, some $\mathbf{w}_f$ are empirically determined for the PUMA-560 and two types of camera models. To examine the causes of combined model equi-output transformation symmetry groups, let us first define the main equi-output transformation groups in combined models.

Let $\mathbb{DH}_m$ be the set of $m \times 4$ finite dimensional matrices with real elements. Let DH be a group defined on $(\mathbb{DH}_m, +)$ with the real + operator. Let $\mathsf{dh}_a, \mathsf{dh}_c \in \mathsf{DH}$. Select $\mathfrak{J}^n, \mathfrak{C}$. Let $\boldsymbol{\theta}_i$ be any element of $\mathfrak{J}^n$. Let $\mathbf{x}_a = \mathcal{F}(\boldsymbol{\theta}_i)|\mathsf{dh}_a$ be a forward kinematic function defined on $\boldsymbol{\theta}_i$ with DH parameters dh_a. Let $\mathbf{x}_a$ be the result of an homogeneous transformation $\mathbf{A}_a$ of $\mathbf{x}_a$. Let $\mathbf{P}_a$ be a camera projection of $\mathbf{x}_a$ onto a stereo-image space $\mathbf{x}'_a$. Similarly for points $(\mathbf{x}_c, \mathbf{x}_c, \mathbf{x}'_c)$ and transformations $(\mathcal{F}(\boldsymbol{\theta}_i)|\mathsf{dh}_c, \mathbf{A}_c, \mathbf{P}_c)$. A representation of the combined camera manipulator kinematic model equi-output transformation symmetry groups are presented in the following commutative diagram

$$\begin{array}{ccccccc}
\boldsymbol{\theta}_i & \xrightarrow{\mathcal{F}(\boldsymbol{\theta}_i)|\mathsf{dh}_a} & \mathbf{x}_a & \xrightarrow{\mathbf{A}_a} & \mathbf{x}_a & \xrightarrow{\mathbf{P}_a} & \mathbf{x}'_a \\
\mathbf{I}\downarrow & & \mathbf{T}\downarrow & & \uparrow\mathbf{E} & & \downarrow\mathbf{I} \\
\boldsymbol{\theta}_i & \xrightarrow{\mathcal{F}(\boldsymbol{\theta}_i)|\mathsf{dh}_c} & \mathbf{x}_c & \xrightarrow{\mathbf{A}_c} & \mathbf{x}_c & \xrightarrow{\mathbf{P}_c} & \mathbf{x}'_c
\end{array} \tag{4.9}$$

where operators over horizontal lines and adjacent to vertical map points between coordinatisations of spaces. In the combined camera manipulator kinematic system there exist equi-output transformation groups of model parameters $(\mathsf{dh}_b, \mathbf{A}_b, \mathbf{P}_b)$ where $\mathsf{dh}_c = \mathsf{dh}_b + \mathsf{dh}_a$, $\mathbf{A}_c = \mathbf{A}_b \mathbf{A}_a$, $\mathbf{P}_c = \mathbf{P}_b \mathbf{P}_a$.

These transformations of combined model parameters induce equivalent transformations in Cartesian $\mathbf{x}_c = \mathbf{T}\mathbf{x}_a$, and image ${}^{\backslash}\mathbf{x}_c = \mathbf{T}^{\backslash}\mathbf{x}_a$ spaces that result in output invariance between $\mathbf{x}'_a$ and $\mathbf{x}'_c$. $\mathbf{x}'_a = \mathbf{x}'_c$, outputs of the combined models are equivalent under the model symmetry transformation induced by $\mathsf{dh}_b, \mathbf{A}_b, \mathbf{P}_b$ if these belong to the combined model equi-output symmetry group. Inspection of (4.9) reveals that the combined model symmetry groups have the structure of a continuous manifold since it is possible to define an infinitesimal change $\mathsf{dh}_b, \mathbf{A}_b, \mathbf{P}_b$. See (4.8). The equi-output transformation groups of combined model parameters depend on the specific kinematics of robot and camera models. Because of the expression complexity of the forward kinematic transformation, it is difficult to explicitly derive closed form expressions for all the equi-output transformation groups for a given robot kinematic geometry and stereo camera model represented in some coordinate systems. It is also difficult to determine all of the subsets of the combined model which when fixed induce a null equi-output transformation group on the complement parameter sets. This is the condition that must be established to define simultaneous absolute calibration of combined camera and kinematic models. Let the two stereo-camera model parameters be arranged in a vector c. It is required to determine $\{\mathsf{dh}^* \subset \mathsf{DH}\} \bigcup \{\mathbf{c}^* \subset \mathbf{c}\}$ which, if fixed, constrain $\mathsf{dh}_b = 0, \mathbf{A}_b = \mathbf{I}, \mathbf{P}_b = \mathbf{I}$ in (4.9).

In the remainder of this Chapter $\mathsf{dh}^*, \mathbf{c}^*$ are determined for a PUMA-560 and two stereo-camera models by a numerical experimentation procedure. Given correct $\mathcal{F}_1^s$ and decalibrated $\mathcal{F}_2^s$ combined stereo-camera and kinematic models, $\mathsf{dh}^*, \mathbf{c}^*$ are determined by the following algorithm:

1. generate a recalibration data set with $\mathcal{F}_1^s$,
2. fix an arbitrary first or an arbitrary additional parameter in the combined camera-manipulator kinematic system,
3. minimise the calibration error (4.1),
4. check if the parameters of the recalibrated $\mathcal{F}_2^s$ equal those of $\mathcal{F}_1^s$:
   (a) if not, then this indicates that not enough parameters in the combined camera-kinematic system have been fixed. Return to step 2.
   (b) if so, then the combined-camera kinematic calibration system is hypothesised to be well defined. Go to step 1, re-initialisation of the data generator $\mathcal{F}_1^s$ and the model $\mathcal{F}_2^s$ and return to step 3 to confirm the hypothesis. If the hypothesis is confirmed 10 times then it is probable that this set of fixed $\mathsf{dh}^*, \mathbf{c}^*$ removes the equi-output transformation symmetry from the remainder of the DH and camera model parameters.

To determine a highly probably desecription of the complete null equi-output combined model parameter group, the above algorithm must be repeated for all possible null group parameter combinations.

4.3 Simultaneous Calibration of the Composition of Camera and DH Models

In order to minimise (4.1), it is necessary to compute error gradients of the image measurements with respect to the combined model parameters w_{jk}. Let $\mathbf{A}_1$ be a transform that maps points from $\mathbf{x}$ to camera space ${}_1\mathbf{x}$ for camera one. Let $\mathbf{A}_2$ be a transform to map points from $\mathbf{x} \in \mathbb{C}$ to camera space

${}_2\mathbf{x}$ for camera two ${}_1\mathbf{x} = \mathbf{A}_1\ \mathbf{x}$. This can be expanded to

$$\begin{bmatrix} {}_1x \\ {}_1y \\ {}_1z \\ 1 \end{bmatrix} = \mathbf{A}_1 \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{4.10}$$

The projection operator

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}, \tag{4.11}$$

maps ${}_1\mathbf{x}$ to image points ${}_1\mathbf{x}'$ such that

$$\mathbf{x}' = \mathbf{P}\mathbf{A}_1\,{}_1\mathbf{x}, \tag{4.12}$$

with

$${}_1x' = \frac{{}_1xf}{{}_1z}, \tag{4.13}$$

$${}_1y' = \frac{{}_1yf}{{}_1z}, \tag{4.14}$$

$${}_1\mathbf{x}' = \begin{bmatrix} {}_1x' & {}_1y' \end{bmatrix}^T. \tag{4.15}$$

If model parameters are organised as a vector $\mathbf{w} = w_{jk}$ where j indexes $\mathbf{T}_i$ and $\mathbf{A}_{1,2}$ and k indexes internal parameters of $\mathbf{T}_i, \mathbf{A}_{1,2}$, then the calibration error gradient is

$$\frac{\partial E}{\partial w_{jk}} = \sum_{i=1}^{n} (\mathbf{P}\mathbf{A}(\boldsymbol{\Gamma}(\boldsymbol{\theta}_i) - \mathbf{x}_i') \frac{\partial \mathbf{P}\mathbf{A}(\boldsymbol{\Gamma}(\boldsymbol{\theta}_i))}{\partial w_{jk}}. \tag{4.16}$$

(4.16) can be evaluated for instance for the x image coordinate of camera 1 by

$$\frac{\partial {}_1x'}{\partial w_{jk}} = \frac{\partial \frac{{}_1xf}{{}_1z}}{\partial w_{jk}} \tag{4.17}$$

$$= \frac{{}_1z \frac{\partial {}_1xf}{\partial w_{jk}} - {}_1xf \frac{\partial {}_1z}{\partial w_{jk}}}{({}_1z)^2}, \tag{4.18}$$

and similarly for ${}_1y', {}_2x', {}_2y'$. See §3.3 for a definition of the partial derivatives of homogeneous transforms with respect to DH parameters. Focal length f can also be a calibration parameter by

$$\frac{\partial {}_1x'}{\partial f} = \frac{{}_1x}{{}_1z}. \tag{4.19}$$

It would be possible to apply similar logic to that of §3.3 to derivation of a second order exact inverse Hessian minimisation of combined model calibration error minimisation. However in the general case of combined n camera calibration, a partial error Hessian has $2(n+1)^2$ sub blocks for camera interactions where n is the number of cameras. Each of these sub-blocks requires taking appropriate second derivatives of (4.18). These cumbersome evaluations are avoided with a quasi-Newton nonlinear

optimisation method, for instance BFGS. See (Press et al., 1992). Hence BFGS is used to minimise (4.1). BFGS iteratively builds an approximation $\mathbf{G}$ to $\mathbf{H}^{-1}$. If

$$\mathbf{p} = \mathbf{w}_{t+1} - \mathbf{w}_t, \tag{4.20}$$

$$\mathbf{v} = \frac{dE}{d\mathbf{w}_{t+1}} - \frac{dE}{d\mathbf{w}_t}, \tag{4.21}$$

$$\mathbf{u} = \frac{\mathbf{p}}{\mathbf{p}^T\mathbf{v}} - \frac{\mathbf{G}_t\mathbf{v}}{\mathbf{v}^T\mathbf{G}_t\mathbf{v}}, \tag{4.22}$$

then BFGS inverse Hessian update at iteration $t+1$ is

$$\mathbf{G}_{t+1} = \mathbf{G}_t + \frac{\mathbf{p}\mathbf{p}^T}{\mathbf{p}^T\mathbf{v}} - \frac{(\mathbf{G}_t\mathbf{v})\mathbf{v}^T\mathbf{G}_t}{\mathbf{v}^T\mathbf{G}_t\mathbf{v}} + (\mathbf{v}^T\mathbf{G}\mathbf{v})\mathbf{u}\mathbf{u}^T. \tag{4.23}$$

If $\mathbf{G}$ is initialised as symmetric positive definite, e.g. the identity matrix, (4.23) avoids Hessian rank loss inversion problems of exact Newton direction minimisation. (4.23) is also more computationally efficient than computation of the partial error Hessian followed by full evaluation of the error Hessian and its inversion. In order to avoid the full Newton step of (3.29) taking the search beyond the range of validity of a quadratic error approximation, BFGS requires a line search algorithm to determine the step size α of descent in the approximate second order minimisation direction

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\mathbf{G}\frac{dE}{d\mathbf{w}}. \tag{4.24}$$

Measurements of the position of a single feature on a manipulator in the two image spaces can be used to calibrate DH parameters determining position in $\mathbb{R}^3$. Examination of forward kinematic equations of position of spatial manipulators with orientation fixed reveal that the DH parameter α_3 cannot be identified since it is not present in the forward kinematic positioning equations. The forward kinematic positioning equations of a 3-$\mathbf{R}$ manipulator are

$$\mathbf{T} = \prod_{j=1}^{3}\mathbf{T}_j = \begin{bmatrix} a & b & c & x \\ e & f & g & y \\ i & j & k & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.25}$$

which when expanded for the position elements

$$\mathbf{x} = \mathbf{T}_{i,4} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T, \tag{4.26}$$

result in the following expressions for x, y, z

$$\begin{aligned} x = {} & a_3 \cos\theta_3 \cos\theta_1 \cos\theta_2 - a_3 \cos\theta_3 \sin\theta_1 \cos\alpha_1 \sin\theta_2 - a_3 \sin\theta_3 \cos\theta_1 \sin\theta_2 \cos\alpha_2 \\ & - a_3 \sin\theta_3 \sin\theta_1 \cos\alpha_1 \cos\theta_2 \cos\alpha_2 + a_3 \sin\theta_3 \sin\theta_1 \sin\alpha_1 \sin\alpha_2 + d_3 \cos\theta_1 \sin\theta_2 \sin\alpha_2 \\ & + d_3 \sin\theta_1 \cos\alpha_1 \cos\theta_2 \sin\alpha_2 + d_3 \sin\theta_1 \sin\alpha_1 \cos\alpha_2 + \cos\theta_1 a_2 \cos\theta_2 - \sin\theta_1 \cos\alpha_1 a_2 \sin\theta_2 \\ & + \sin\theta_1 \sin\alpha_1 d_2 + a_1 \cos\theta_1, \end{aligned} \tag{4.27}$$

$$\begin{aligned} y = {} & a_3 \cos\theta_3 \sin\theta_1 \cos\theta_2 + a_3 \cos\theta_3 \cos\theta_1 \cos\alpha_1 \sin\theta_2 \\ & - a_3 \sin\theta_3 \sin\theta_1 \sin\theta_2 \cos\alpha_2 + a_3 \sin\theta_3 \cos\theta_1 \cos\alpha_1 \cos\theta_2 \cos\alpha_2 \\ & - a_3 \sin\theta_3 \cos\theta_1 \sin\alpha_1 \sin\alpha_2 + d_3 \sin\theta_1 \sin\theta_2 \sin\alpha_2 \\ & - d_3 \cos\theta_1 \cos\alpha_1 \cos\theta_2 \sin\alpha_2 - d_3 \cos\theta_1 \sin\alpha_1 \cos\alpha_2 \\ & + \sin\theta_1 a_2 \cos\theta_2 + \cos\theta_1 \cos\alpha_1 a_2 \sin\theta_2 - \cos\theta_1 \sin\alpha_1 d_2 + a_1 \sin\theta_1, \end{aligned} \tag{4.28}$$

$$\begin{aligned} z = {} & \sin\alpha_1 \sin\theta_2 a_3 \cos\theta_3 + a_3 \sin\theta_3 \sin\alpha_1 \cos\theta_2 \cos\alpha_2 + a_3 \sin\theta_3 \cos\alpha_1 \sin\alpha_2 \\ & - d_3 \sin\alpha_1 \cos\theta_2 \sin\alpha_2 + d_3 \cos\alpha_1 \cos\alpha_2 + \sin\alpha_1 a_2 \sin\theta_2 + \cos\alpha_1 d_2 + d_1. \end{aligned} \tag{4.29}$$

x, y, z do not contain α_3. α_3 contributes to elements of the upper left 3×3 orientation matrix of $\mathbf{T}$ only. Hence α_3 cannot be calibrated from positional measurements of x, y, z.

After positioning DH parameters have been calibrated, orientation DH parameters in $\prod_{i=3}^{6} \mathbf{T}_i$ and α_3 can be re-calibrated. From Table 3.1, with stereo-cameras, two additional features are required with $\mathbf{T}_f^2, \mathbf{T}_f^3$ known a priori. Camera one measurements ${}_1^1\mathbf{x}', {}_1^2\mathbf{x}', {}_1^3\mathbf{x}'$ and camera two measurements ${}_2^1\mathbf{x}', {}_2^2\mathbf{x}', {}_2^3\mathbf{x}'$ can be mapped via $\mathbf{S}$ to ${}^1\mathbf{x}, {}^2\mathbf{x}, {}^3\mathbf{x}$. Given, ${}^1\mathbf{x}, {}^2\mathbf{x}, {}^3\mathbf{x}$ and $\mathbf{T}_f^1, \mathbf{T}_f^2, \mathbf{T}_f^3$, $\mathbf{x}_i$ the position in $\mathbb{C}$ of a manipulator according to stereo-camera measurements can be computed. DH parameters governing manipulator orientation can be calibrated by gathering i image measurements of manipulator position $\mathbf{x}_i' = {}_1^1\mathbf{x}', {}_1^2\mathbf{x}', {}_1^3\mathbf{x}', {}_2^1\mathbf{x}', {}_2^2\mathbf{x}', {}_2^3\mathbf{x}'$ and mapping these to $\mathbf{x}_i$. Hence then can be used to re-calibrate using the method of §3.3.

4.3.1 A Calibration Experiment with a Homogeneous Camera Model

Let $\mathbf{A}_{1,2}$ be homogeneous transforms. 34 numerical experiments were undertaken to determine a set of fixed parameters that induced no equi-output transformation structure in the remaining parameters of a $\prod_{i=1}^{3} \mathbf{T}_i\mathbf{A}$ combined model. A more complete combinatorial experiment design was not undertaken due to time and computing limitations. It was decided to simplify numerical experiment search by initially fixing all orientation parameters $\prod_{i=4}^{6} \mathbf{T}_i$. A subset of DH and stereo-camera parameters that can be absolutely calibrated simultaneously was arrived at by empirical numerical experimentation for the remaining $\prod_{i=1}^{3} \mathbf{T}_i$ and $\mathbf{A}_1, \mathbf{A}_2$. Since f is generally known and reliable, it was assumed to be correct. For the interior camera model $\mathbf{P}$, a negative focal length $-f$ is required to rectify the pin-hole projection to correspond to right-way up images delivered by image processing boards.

The combined DH and camera models must be initialised so that the z coordinate of the projection to camera image space is positive. Otherwise a calibration solution is generated for a reflection of the camera plane. With a division by z in the region -1 to 0, as z approaches zero, coordinates $x/z, y/z$ go to $-\infty$, until 0 is crossed. Then x, y coordinates come down from $+\infty$. $z > 0$ is also a constraint during minimisation to avoid entrapment in local minima. This constraint can be difficult to detect since it depends on the relative positions of camera and manipulator.

Camera-DH parameter interaction is complex. Numerical experiments indicate that if no knowledge of the camera system is assumed, then only 2 DH parameters, for instance a_2, a_3, can be unknown

in order for the equi-output parameter group to collapse to the null group. If it is assumed that $\mathbf{T}_3$ and α_{c1}, α_{c2} are known, the remaining parameters for $\mathbf{T}_2, \mathbf{T}_3$ and camera1 and camera2 can be calibrated. This is a total of 6 DH parameters and 6 combined homogeneous camera parameters. Since $\mathbb{R}^4$ stereo camera measurements are an embedding of a $\mathbb{R}^3$ manifold, each sample generates 3 pieces of information. Given 12 combined model parameters, at least 4 samples that span embedding space $\mathbb{R}^4$ are required to identify model parameters. Figure 4.2 shows the results of one of the experiments for combined absolute calibration of manipulator kinematic and camera model parameters. The PUMA 560 of Table 2.1 is de-calibrated for $\mathbf{T}_1, \mathbf{T}_2$ by a perturbation by a uniform random distribution of 5 cm to Table 4.1 The reference models and initialisations are:

link	joint	twist	length	offset
i	θ_i(rad)	$\alpha_i(rad)$	a_i(m)	d_i (m)
1	θ_1	1.596	0.016	0.0045
2	θ_2	0.0258	0.4811	0.0474
3	θ_3	-1.5708	0.0203	0.1254

Figure 4.1: A De-calibrated PUMA 560 DH Table.

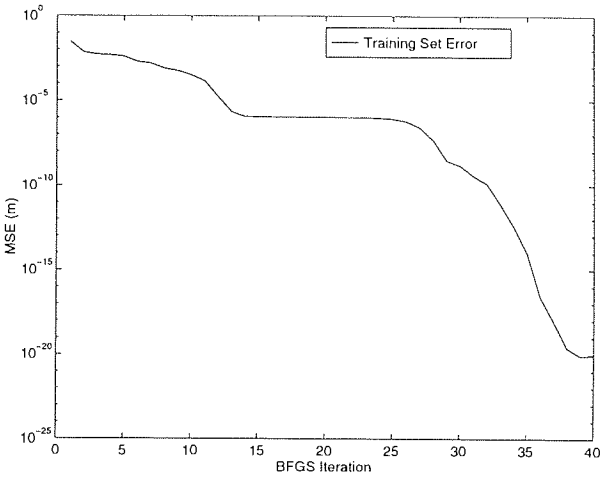

Figure 4.2: Calibration of a Composition of DH and Homogeneous Camera Parameters.

1. Reference camera 1 is $\theta = 0.9, a = -1.4, \alpha = -\pi/2, d = 1.3$.
2. Model camera 1 model is initialised as $\theta = 0.8, a = -1.3, \alpha = -\pi/2, d = 1.4$.
3. Reference camera 2 is $\theta = 1.6, a = -1.5, \alpha = -\pi/2, d = 1.1$.
4. Model camera 2 model is initialised as $\theta = 1.5, a = -1.6, \alpha = -\pi/2, d = 1.2$.
5. The algorithm convergence criteria are that $SSE < 10^{-6}m^2$ and $\frac{dE}{dw_k} < 10^{-7}m^2 \forall w_k \in \mathbf{w}$

BFGS was implemented in Matlab with cubic interpolation line search (see (Press et al., 1992), pg. 384-386). Figure 4.2 shows that $MSE = 10^{-20}m^2$ after 38 BFGS iterations. The shape of the minimisation graph contains two error plateaus. One plateau occurs after approximately 15 BFGS iterations at approx $MSE = 10^{-5}m^2$, i.e. 5mm, and continues until about 25 iterations. The second plateau begins after approximately 37 iterations at $MSE = 10^{-20}m^2$, i.e. at an average precision of 0.00000001 mm. BFGS is sensitive to variable scaling and rounding errors. The use of a cubic line search can also introduce artifacts due to the incorrect approximation of stiff cubic polynomials of a quasi-quadratic minimisation surface and sensitivity of cubics to round off error. The cause of the first minimisation error plateau is that several of the variables are close to minimised in the first 10 iteration. This causes a variable scaling problem. This leads to round-off error in the convergence of the BFGS approximation of the inverse Hessian. The final high level of numerical precision is a little misleading since machine fabrication tolerances are generally not higher than 1mm which several

times more accurate that the measurement precision of stereo-cameras, generally of the order of $1cm$. See §6.1.2.1.

4.3.2 A Calibration Experiment with a Pan-Tilt Camera Model

A camera space can have 3 *dof* translational and 3 *dof* rotational difference to Cartesian space. Since the general case of arbitrary positional and rotational displacement from Cartesian space means less manipulator DH parameters can be calibrated, it is useful to impose environmental constraints on camera model parameter values. Camera roll can be constrained to zero by a gravitational device such as a spirit level. Assuming that we can thus neglect camera roll by constraining it to be zero with respect to a Cartesian coordinate system, if a camera is mounted on a pan-tilt platform, let $[X, Y, Z, 1]$ be a vector representing positional displacement of the origin of the image plane from Cartesian origin. If pan ϕ is the angle between the x and $`x$ axes and tilt $\alpha$ is the angle between $z$ and $`z$ axes, then an exterior model of a pan-tilt camera is

$$\mathbf{R} = \begin{bmatrix} \cos\phi & \sin\phi & 0 & X \\ -\sin\phi\cos\alpha & \cos\phi\cos\alpha & \sin\alpha & Y \\ \sin\phi\sin\alpha & \cos\phi\sin\alpha & \cos\alpha & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.30}$$

and $`\mathbf{x} = \mathbf{PRx}$. See (Fu et al., 1987). This more realistic (than a homogeneous) camera model was tested on re-calibration of PUMA 560 in Table 4.3 to Table 2.1. The PUMA 560 of Table 2.1 is de-calibrated for $\mathbf{T}_1, \mathbf{T}_2$ by perturbation by a uniform random distribution of 5 cm. If f is fixed, 35 numerical simulations indicate that with the (Fu et al., 1987) camera model, 4 DH parameters can be absolutely calibrated simultaneously with the camera parameters. Figure 4.4 shows the BFGS error convergence curve for a representative trial where there are four variable DH parameters, all variable $\mathbf{R}$ parameters, and fixed f parameter where:

1. focal length is $f = 5.6mm$.
2. Reference camera 1 is $\theta = -\pi/2, X = 0.5m, Y = 0.5m, Z = 1.0$.
3. Model camera1 is initialised as $\theta = -\pi/3, X = 0.6m, Y = -0.4m, Z = 1.1m$.
4. Reference camera 2 is $\theta = -\pi/2, X = 0.2, Y = -0.1, Z = 1.2$.
5. Model Camera2 is initialised as $\theta = -\pi/1.5, X = 0.3, Y = -0.2, Z = 1.3$.
6. Algorithm convergence criteria are $SSE < 10^{-6}m^2$ and $\frac{dE}{dw_k} < 10^{-7}m^2 \forall w_k \in \mathbf{w}$, the non-fixed combined model parameters.

In Figure 4.4 there are two error minimisation plateaus at $MSE = 10^{-5}m^2$ and $MSE = 10^{-20}m^2$ These plateaus are similar to the plateaus in Figure 4.2. These plateaus are also a result of the sensitivity of BFGS to rounding error and variable scaling.

link	joint	twist	length	offset
i	θ_i(rad)	$\alpha_i(rad)$	a_i(m)	d_i (m)
1	θ_1	1.6175	0.026	0.0
2	θ_2	0.0192	0.4733	0.0
3	θ_3	-1.5708	0.0203	0.1254

Figure 4.3: A De-calibrated PUMA 560 Denavit Hartenberg Table.

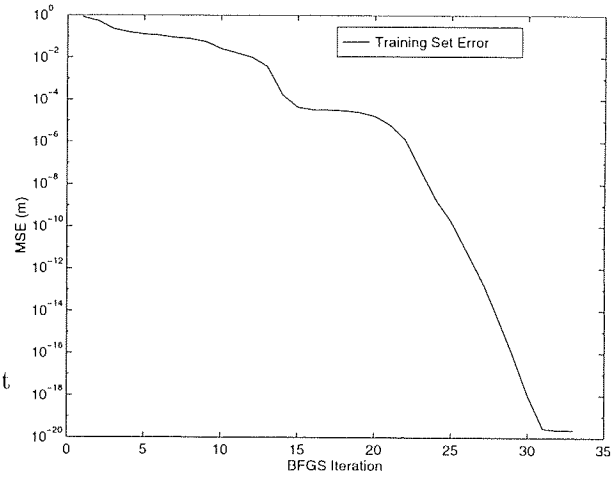

Figure 4.4: Calibration of a Composition of DH and Pan-Tilt Camera Parameters.

Chapter 5

Nonparametric Regression on Kinematic Functions

Nonparametric models are necessary to model and explain data if analytical explanatory models cannot be derived. There are many interesting nonparametric regression and function approximation models with various properties, for instance flexible discriminants, kernel regression, classification and regression trees, multi-variate adaptive regression splines (Friedman, 1991), ridge regression, projection pursuit regression, radial basis functions and feed-forward networks. See (Rohwer, 1994; Bishop, 1995; Ripley, 1996; Seber and Wild, 1989). This work mostly considers feed-forward networks and radial basis functions.

Nonparametric models may replace parametric models of manipulator kinematics. This Chapter analyses proposals by (Miller, 1989; Miller et al., 1990; Kuperstein and Rubinstein, 1989; Cooperstock and Milios, 1993; Mel, 1990; Ritter et al., 1992; Walter and Schulten, 1993; Jordan and Rumelhart, 1992; Barhen et al., 1989; Kuperstein and Rubinstein, 1989; DeMers, 1993; Kawato et al., 1987; van der Smagt, 1994; Edelman, 1989; Eckmiller, 1989; Hashimoto et al., 1992) for nonparametric representations of inverse kinematic solutions of non-redundant and redundant manipulators. Some of the previous nonparametric control system developers argue that nonparametric approximations of parametric manipulator kinematic and stereo-camera models may have advantages, for instance real-time efficiency, over parametric models. These claims are evaluated in this Chapter as part of a general analysis of the role of nonparametric regression in robotics and automation systems.

To illustrate the deployment of nonparametric regression in a robotic system, an experiment is undertaken to determine a nonparametric model correction of nonlinear auto-focus, depth of field, and radial lens distortion calibration error residuals remaining after calibration of an affine stereo model to Cartesian space of a physical manipulator gripping a red LED. The nonparametric corrector's model complexity and parameters are optimally determined by hierarchical cross-validation and the affine stereo model parameters are also determined by cross validation of a singular valued decomposition determination of affine stereo model parameters.

5.1 Feed-forward Networks and Radial Basis Functions

Radial basis functions and feed-forward networks are powerful nonparametric models since they can uniformly approximate arbitrary continuous functions defined over compacta (Bishop, 1995). Feed-forward and radial basis function regression is usually defined on an arbitrarily sized data set $\mathbb{D} =$

$\{\boldsymbol{\theta}_i, \mathbf{x}_i\}$ generated by an underlying stochastic regression function $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^o : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$. Inputs $\boldsymbol{\theta}_i$ of $\mathbb{D}$ are independently drawn from a distribution $P(\boldsymbol{\theta}_i)$ and targets $\mathbf{x}_i$ are generated by a mapping $\mathcal{F}$ to which is added a random variable ϖ (usually assumed to be a zero mean Gaussian distribution)

$$\mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i) + \varpi_i. \tag{5.1}$$

In order to use a nonparametric model to approximate $\mathcal{F}$, it is necessary first to specify a regression error measure. Error measures L_x can be designed to satisfy various criteria, e.g. SSE or minimax (Bishop, 1995). This work usually considers the sum of squares error (SSE) measure. If $F(\boldsymbol{\theta})$ represents (5.3) then

$$SSE = \sum_{i=1}^{n} (F(\boldsymbol{\theta}_i) - \mathbf{x}_i)^2 \tag{5.2}$$

for n elements of a data set. In order to fit the data set, it is necessary to minimise an error measure such as SSE after selection of a regression model such as a feed-forward network or radial basis functions.

5.1 Definition (The Class of Feed-forward Networks)
A feed-forward network $F : \mathbb{R}^n \rightarrow \mathbb{R}^o$ is defined by

$$x_k^i = f_k(\sum_{j=0}^{m} w_{jk} f_j(\sum_{i=0}^{n} w_{ij}\theta_i)) \text{ for } k = 1, \ldots o. \tag{5.3}$$

(5.3) has an input layer containing $n+1$ elementary functions, a hidden layer containing $m+1$ elementary functions and an output layer containing o elementary functions. θ_0 and f_0 are bias functions which always output unity. Output function x_k^i are indexed by $k = 1, \ldots, o$ for datum i. f_j is usually a squashing function such as a sigmoid or a tanh. f_k may be squashing or linear functions to satisfy various error function design criteria (Bishop, 1995). If certain conditions related to complexity of the underlying function to be approximated and cardinality of elementary functions are met, a feed-forward network is a universal function approximator. See (Ripley, 1996), pg. 173-178.

(Bishop, 1995) analyses the **curse of dimensionality** (Bellman, 1961) of nonparametric regression. The curse of dimensionality is that in general the number of samples required for well specified regression grows exponentially with the dimensionality d of the regression input space. If each variable is discretised into M measurements, scaling of regression is of the order of M^d. See (Bishop, 1995; Ripley, 1996). In certain cases, feed-forward networks more efficiently approximate some classes of functions under different sets of function norms compared to fundamental universal function approximators such as Fourier expansions or orthogonal polynomials. Feed-forward networks may scale better on some higher dimensional regression problems compared to for instance some classes of orthogonal polynomials. If the number of hidden functions of a feed-forward network is greater (lesser) than the number of input functions, the feed-forward network projects the input coordinate system to a higher (lower) dimensional space by an affine transformation followed by the nonlinear, e.g. sigmoidal, transformation. The function to be approximated may be more economically represented, i.e. require fewer affine parameters, by the affine transformation to the output space compared to an orthogonal polynomial regression.

Since (5.3) is nonlinear, determination of parameters that minimise a regression error measure is a nonlinear optimisation problem defined on parameters $\{w_{ij}, w_{jk}\}$ of F. If the parameters $\mathbf{w}_{ij}$ are organised as a vector $\mathbf{w}_0$ at iteration 0, a SSE minimisation algorithm at each iteration t computes

a line direction $\mathbf{u}_t$ and a step size η_t in order to update $\mathbf{w}$ by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{u}_t. \tag{5.4}$$

First order minimisation direction is generally estimated by $\mathbf{u}_t^i = -\frac{dSSE}{dw_i}$, the negative error gradient. Step size η_t can be fixed or modified by error momentum terms or determined by a line search. Minimisation of (5.4) in the $\mathbf{u}_t$ direction can be corrected to second order direction by Newton or quasi Newton direction terms. See §3.29 and (Seber and Wild, 1989; Bishop, 1995; Ripley, 1996). Iterative minimisation of (5.4) in the first order direction of $\mathbf{u}_t$ or a second order direction reduces $E(\mathbf{w}_t)$ to a local E_l^* or global $E_g^{\textcircled{w}}$ minimum. $E_g^{\textcircled{w}} \leq E_l^*$, $\forall g, l$ where l indexes local minima and g indexes global minima. Feed-forward network parameter $\mathbf{w}$ space is a representation of the dihedral group of the k-dimensional cube (Chen et al., 1993). For a fully connected feed-forward network with M hidden functions, the set of equi-output transformations g of $\mathbf{w}_a$, $\mathbf{w}_b = g(\mathbf{w}_a)$ of $\mathbf{w}$ can be factored into two sub-groups $\mathbb{IN} \times \mathbb{FL}$. $\mathbb{IN}$ are the $M!$ hidden function parameter 'interchange' transformations and $\mathbb{FL}$ are the 2^M sign flip parameter transformations. There are thus at least $M!2^M$ global minima solutions for a quadratic SEE regression on a feed-forward network.

Since expansions of non-orthogonal basis[1] functions may scale better than orthogonal basis functions, a second useful class of nonparametric models is the class of non-orthogonal radial basis functions.

5.2 Definition (The Class of Radial Basis Functions)
An expansion[2] of radial basis functions $F : \mathbb{R}^n \rightarrow \mathbb{R}^o : \mathbf{x}_i = F(\boldsymbol{\theta}_i)$ is a linear combination of k generally non-orthogonal nonlinear basis functions $\phi_j, 1 \leq j \leq k$ such that

$$x_k^i = \sum_{j=0}^{k} w_{kj} \phi_j(\boldsymbol{\theta}_i) \tag{5.5}$$

for data $1 < i < z$. Biases are $w_{k0}\phi_0$ where $\phi_0 = 1$.

See (Lowe, 1995).

Radial basis functions ϕ_k generally have a centre $\mathbf{c}_k$. ϕ_j can be bounded or unbounded. Bounded ϕ_j can have compact or non-compact support. Examples of ϕ_k are listed in Table 5.1. Since unbounded ϕ_j have better interpolating properties than bounded ϕ_j (Lowe, 1995), Gaussian kernels are used in this work. A Gaussian basis function ϕ_j has an associated centre $\mathbf{c}_j$ and variance $\boldsymbol{\sigma}_j$. With Gaussian basis functions, (5.5) becomes

Gaussian: $\phi_k(\boldsymbol{\theta}_i) = \exp(\frac{-(\|\boldsymbol{\theta}_i - \mathbf{c}_k)^2\|}{2\sigma_k^2})$
Multi-quadric: $\phi_k(\boldsymbol{\theta}_i) = (\boldsymbol{\theta}_i^2 + \mathbf{c}^2)^{\frac{1}{2}}$
Transcendental: $\phi_k(\boldsymbol{\theta}_i) = \cos^2(\boldsymbol{\theta}_i - \mathbf{c}_k)$
Thin plate spline: $\phi_k(\boldsymbol{\theta}_i) = \boldsymbol{\theta}_i^2 log(\boldsymbol{\theta}_i)$
Cauchy: $\phi_k(\boldsymbol{\theta}_i) = \frac{1}{1+(\boldsymbol{\theta}_i - \mathbf{c}_k)^2}$

Table 5.1: Classes of Radial Basis Functions.

$$x_k^i = \sum_{j=0}^{k} w_{kj} \exp(\frac{-(\|\boldsymbol{\theta}_i - \mathbf{c}_j)^2\|}{2\sigma_j^2}). \tag{5.6}$$

Since Gaussian basis functions are continuous, bounded, and have finite non-zero integral, expansions of Gaussian basis functions can uniformly approximate continuous functions on compacta. See (Ripley, 1996), pg. 132. For uniform approximation of arbitrary continuous functions, the number of required radial basis functions scales $O(b\bar{n}^d)$, where d is the dimensionality of the function approximation input space, n is the average number of basis functions per dimension for uniform approximation $\leq \epsilon$ and b is the dimensionality of the output space.

(Bishop, 1995) describes several methods to minimise the SSE of an expansion of Gaussian basis functions fit to a data set. A straightforward approach is to determine $\mathbf{c}_j$ by k-means based clustering

[1] Radial basis functions are sometimes called kernel functions.

[2] Radial basis functions can also be derived as a Green's function approximation of a regularised error functional solution of some general classes of differential equations. See (Ripley, 1996) pg. 137. and (Bishop, 1995).

algorithms possibly with pruning of means, combined with input data Voronoi tessellation based determination of σ_j^2. Once $\mathbf{c}_j, \boldsymbol{\sigma}_j$ are determined, the radial basis functions becomes a generalised[3] linear regression model. Hence w_{kj} can be determined by pseudoinversion

$$\mathbf{W}^* = \mathbf{X}\Phi^+, \tag{5.7}$$

where $\mathbf{X} = [\mathbf{x}_1 \vdots \mathbf{x}_z]$ and $\Phi = \phi_j^i$. The main limitation of heuristic clustering of basis function parameters to generate a radial basis function approximation of an unconditional density function is is a sub-optimal representation of conditional regression density. Nonlinear optimisation of all radial basis function parameters $(\mathbf{c}_j, \sigma_j^2, w_{kj})$ provides a better estimate of conditional regression density but this is more computationally intensive. See (Bishop, 1995).

5.2 Minimisation of the Bias and Variance of a Nonparametric Regression

With an assumption of Gaussian noise, minimisation of the SSE of a nonparametric model fit to a data set maximises the likelihood of model fit to data since

$$E_{\mathbb{D}} = -\log(P(\mathbb{D}|\mathbf{w})), \tag{5.8}$$

and the likelihood is

$$P(\mathbb{D}|\mathbf{w}) = \exp(-E_{\mathbb{D}}). \tag{5.9}$$

The maximum likelihood estimate of a regression function underlying a data set is in general a poor and arbitrary estimate of the underlying function. For a given data set there are infinitely many functions which reduce maximum likelihood of model fit to zero given sufficient model complexity. To improve the estimate of the underlying function. It is necessary to employ basic statistical methods for minimisation of estimator bias and variance.

5.3 Definition (The Bias and Variance of a Regressor)
The expected Sum Squared Error $E(SSE)$ may be decoupled into estimator bias and variance error terms (Geman et al., 1992). If $\mathbb{D}_j = \{(\theta_1, x), \ldots, (\theta_k, x_k)\}$, are samples of an underlying distribution $\mathcal{F}$ contaminated for instance by a Gaussian noise process, then for any regression function $F(\theta)$ and any fixed θ

$$E_{\mathbb{D}_j}[F(\theta;\mathbb{D}_j) - E[x|\theta])^2] = (E_{\mathbb{D}_j}[F(\theta;\mathbb{D}_j)] - E[x|\theta])^2 \text{ "bias"} \tag{5.10}$$

$$+ E_{\mathbb{D}_j}[(F(\theta;\mathbb{D}_j) - E_{\mathbb{D}_j}[F(\theta;\mathbb{D}_j)])]^2 \text{ "variance"}. \tag{5.11}$$

If on the average $F(\theta;\mathbb{D}_j)$ is different from $E[x|\theta]$, then $F(\theta;\mathbb{D}_j)$ is a biased estimator of $E[x|\theta]$. The extension of regression bias variance analysis of univariate x to the multi-variate $\mathbf{x}$ case is straightforward. A good estimator F of an underlying distribution $\mathcal{F}$ is one with low variance and low bias. An unbiased estimator may still have a large SSE if the variance is large. Thus either bias or variance can contribute to poor regression estimation. There is a tradeoff between the bias and variance contributions to the estimation error. Typically, variance is reduced by smoothing for instance via

[3] Hence for control applications, once kernel parameters are fixed, observability, reachability, stability, and robustness of radial basis function approximation of a controller can be determined. See definitions 2.14, 2.15, 2.16, 2.17.

combination of influence of nearby samples in input $\boldsymbol{\theta}$ space. This increases bias as details of the regression function are blurred.

A correctly identified parametric model is the optimal minimal bias and variance estimator. If a parametric model cannot be identified, the bias-variance tradeoff involved in nonparametric regression can be controlled by cross validation or regularisation of model complexity in an optimal numerical experimental design. See (Seber and Wild, 1989; Box et al., 1978; Bishop, 1995).

5.2.1 Hierarchical Cross Validation of a Nonparametric Regression

Let $\mathbb{D}$ be partitioned into a multi-model parameter calibration sets V^{-i} and ensemble estimator testing sets V^i. V^{-i} is partitioned[4] into m sets $(V^1, \ldots, V^m)$ (as equally as possible, i.e. a balanced partition). For $i = 1, \ldots, m$, $V^{-i} = \mathbb{D} \backslash V^i$ is an ensemble model second order calibration set and V^i is an ensemble model second order validation set. Let each V^{-i} be split into j equal partitions V_j^{-i}, $j = 1, \ldots, o$. $V_{-j}^{-i} = V_{-i} \backslash V_j^{-i}$ is a first level calibration set and V_j^{-i} is a first level validation set.

5.4 Definition (Hierarchical k-fold Cross Validation)
Let F_z be a set of models where z is[5] a structural complexity indexing parameter. If $\forall j \forall i \forall z, F_z$ is fitted to V_{-j}^{-i} to minimise an error norm $E_{j,i}^z$ (usually L_2, i.e. SSE) over V_{-j}^{-i}, then the average[6] error of F_z on V_j^{-i} over all j is $\bar{E}_i^z = \frac{1}{j} \sum_{j=i}^{o} E_{j,i}$. $z_i^* = \min_{z_i}(\bar{E}_i^z)$ defines the optimal cross validated model $F_{z^*}^i$ for V^i. In the second level of cross validation, r nonparametric models of optimised model structure ${}^rF_{z^*}^i$ for a given V^{-i} are fitted to V^i. The final i ensemble models $\bar{F}_{z^*}^i = \frac{1}{r} \sum_{j=1}^{r} {}^rF_{z^*}^i$ are tested on V_{-i}. $\bar{F}_{z^*}^i$ with the lowest error on V_{-i} is the optimal hierarchically cross validated regressor.

5.2.2 Regularisation of Smoothness of a Nonparametric Regression

Model complexity is a function of the number of basis functions in the model. Regularisation of an error function is a second method to control bias and variance by control of regressor effective complexity

5.5 Definition (Regularisation of Estimator Variance)
With a flexible model, regression variance can be controlled by adding a regularisation term $E_{\mathbf{W}}$, to control smoothness, to $E_{\mathbb{D}}$.

$$E_\Sigma = w^{1*} E_{\mathbb{D}} + w^{2*} E_{\mathbf{W}}, \tag{5.12}$$

where w^{1*} hyper-parameter controls bias and w^{2*} hyper-parameter smoothness of regression. If $E_{\mathbf{W}}$ is quadratic in $\mathbf{x}$, this is known as Tikhonov regularisation (Tikhonov and Arsenin, 1977). See (Bishop, 1995).

Regularisation of model complexity determines model smoothness. Bayesian regression is a theoretical framework for determination of regression optimal w^{1*}, w^{2*} and $E_{\mathbf{W}}$ of nonparametric models (Bishop, 1995). In Bayesian regression

$$p(\mathbf{w}|\mathbb{D}) = \frac{p(\mathbb{D}\mathbf{w})p(\mathbf{w})}{p(\mathbb{D})}. \tag{5.13}$$

[4] Partition of V^{-i} into m sets instead of into a single calibration set and validation set, i.e. single model cross-validation, averages over the effect of randomly choosing a single partition of V^{-i}.

[5] In the case of models with several structural complexity parameters, e.g. radial basis functions which can have even in the simplest case, a one dimensional parameterisation of uniform spherical basis kernels, a parameter indexing the number of basis functions, and a combination of possibilities for random placement of the kernel centres on the data, it is possible to map these three parameters to N in a straightforward manner, e.g after discretisation of possible spherical variance values.

[6] In the simplest case. It is also possible to weight average error in various ways. See (Ripley, 1996).

the posterior regression probability density $p(\mathbf{w}|\mathbb{D})$ is equal to the product of the maximum likelihood of regression $p(\mathbb{D}|\mathbf{w})$ and the prior probability on the regression parameters $p(\mathbf{w})$ normalised by the data probability density, the evidence $p(\mathbb{D})$. If the prior (the probabilistic form of a regularisation function $E_{\mathbf{W}}$) is true, and evaluation of optimal $w^{1*}, w^{2*}, \mathbf{w}$ is computational tractable, the posterior Bayesian regressor is optimal[7] (Bishop, 1995; Ripley, 1996; Seber and Wild, 1989). Definition of a prior may require exploratory data analysis to identify families of regression functions. (5.13) can be extended to selection of optimal models. See (Bishop, 1995). In the high dimensional multi-variate regression, evaluation of optimal $w^{1*}, w^{2*}, \mathbf{w}$ is intractable without approximations, for instance Gaussian, of the posterior regression probability density. See (Bishop, 1995).

5.3 Nonparametric Approximation of Manipulator Functions

Nonparametric regression may approximate functions at different functional levels of robot manipulator control. Table 5.3 groups some proposed applications of nonparametric regression by level of DFA FRM on pg. 24 and task function. See (Bekey and Goldberg, 1991) for variants on these proposed

FRM Level	Functional Task	Citation
❶	Sensor-based Static Inverse Kinematics	(Ritter et al., 1992; Walter and Schulten, 1993) (Kuperstein and Rubinstein, 1989) (Edelman, 1989)
❶	Dextrous Inverse Kinematics	(DeMers, 1993; Jordan and Rumelhart, 1992) (Barhen et al., 1989)
❶+❷	Sensor-based Kino-dynamic	(Kawato et al., 1987; van der Smagt, 1994) (Miller, 1989; Miller et al., 1990)
②+❶	Trajectory/Path Planning	(Mel, 1990)
❶+❷	Acceleration of Image Servoing	(Hashimoto et al., 1992)

Table 5.2: Proposed Applications of Nonparametric Regression to Manipulator Control Problems.

applications of nonparametric regression to manipulator kinematic and dynamic problems. At the robot inverse dynamic control level, nonparametric models have been proposed for approximation of dynamic control components such as friction, stiction, Coriolis and centrifugal forces (Warwick et al., 1992; White and Sofge, 1992; Miller III et al., 1990; Miller, 1989; Miller et al., 1990).

While Table 5.3 classifies proposals for application of nonparametric models to manipulator control problems in DFA functional terms, the proposed applications are usually developed within a behavioural robotics framework. Nonparametric models and associated parameter determination algorithms are usually but not exclusively embedded in behaviourist robot control architectures. In the most general case, behavioural control involves regression by a nonparametric model F of sensor data $\mathbf{x}'_t, \dot{\mathbf{x}}'_t$ onto appropriate effector actions $\boldsymbol{\theta}_{t+1}$ to achieve an operational goal defined on a state space

$$\boldsymbol{\theta}_{t+1} = F(\boldsymbol{\theta}_t, \mathbf{x}'_t, \dot{\mathbf{x}}'_t), \tag{5.14}$$

$$(\mathbf{x}'_{t+1}, \dot{\mathbf{x}}'_{t+1}) = \mathcal{W}(\boldsymbol{\theta}_{t+1}, \mathbf{x}'_t, \dot{\mathbf{x}}'_t), \tag{5.15}$$

[7] Similarly in pattern recognition, a special case of regression on step functions, if a prior on a class classification probability is true and the density of observed classes is approximated correctly and classification loss and risk functions have defined then the posterior classifier is optimal. See (Ripley, 1996).

where $\mathcal{W}$ is a state transition function. See (2.31) and (2.32). (5.14) and (5.15) are time-invariant. Time-variant state space systems $\mathcal{F}_t, \mathcal{W}_t$ may be described with the same equations if $\mathbf{x}'_t$ is expanded to be a vector element of a Markovian embedding space $\mathbf{X}'_t$ of sensor data. In time series prediction terminology, an embedding space $\mathbf{X}'_t$ of sensor data is a space of sufficient dimensionality such that sufficient information is present to make accurate but not necessarily deterministic prediction of $\mathbf{x}'_{t+1}, \dot{\mathbf{x}}'_{t+1}$. Given the noise ϖ inherent in sensor measurements and physical manipulator control, $\mathcal{W}$ is a probability transition matrix which must be deduced a priori or inferred a posteriori from examples of Markov chains generated by robot operation in order to generate correct $\mathcal{F}$. Behavioural design of manipulator controllers involves identification of nonparametric models capable of performing certain tasks followed by estimation of parameters by trial and error operation of a manipulator. Bottom-up composition of nonparametric models increases task complexity.

Nonparametric models may approximate sensor-based task functions with little a priori modelling assumptions. Many proposed applications of nonparametric regression to manipulator control attempt to increases robustness of sensor-servo loop manipulator control, for instance for object approach and grasp followed by insertion or parts mating tasks. Sensor-servo based parts inspection is another commonly proposed application of nonparametric regression solutions. In this case, given a reference inspection image position and orientation in an image coordinate system, if some convenient robust local or global features, e.g. Fourier descriptors, can be extracted from the image, an image inspection task Jacobian $\mathbf{J}$ can be constructed from models of a manipulator $\mathcal{F}$ and sensors $\mathbf{A}$ (Samson et al., 1991).

If position and orientation $\mathbf{x}'_i = \mathbf{A}(\mathcal{F}(\boldsymbol{\theta}_i))$ (see (4.1)) of a manipulator can be identified in sensor space and a goal grasp location and orientation $\mathbf{x}'_g$ can also be specified, then a task function $E = (\mathbf{x}'_i - \mathbf{x}'_g)^2$ can be iteratively minimised if a kinematic task Jacobian $\mathbf{J}_T = \frac{\partial \mathbf{x}'_i}{\partial \theta_j}$ can be evaluated. See §4.3. A task Jacobian is the product of the manipulator Jacobian and the Jacobian of the sensor mapping to Cartesian space. If $\mathbf{J}$ is under-determined, e.g. for a dextrous manipulator, then regularisation functions may be defined. See §2.6. (Espiau et al., 1992) proves that iterative minimisation of E based on $\mathbf{J}_T$ is sufficient for image-based manipulator servo to $\mathbf{x}'_g$. Task Jacobians can also be dynamic $\mathbf{D}_T = \frac{\partial^2 \mathbf{x}'_i}{\partial \theta_k^2}$. See (1.4). (Samson et al., 1991) is a comprehensive analysis of observability, reachability, stability, and robustness of certain classes of servo motions generated by minimisation of certain classes of dynamic task functions by iteration on $\mathbf{D}_T$.

Servoing on a task Jacobian requires several iterations to minimise E. In order to accelerate a inspection task, a feed-forward network $\mathcal{F}$ can approximate an inverse task Jacobian $\mathbf{J}^{-1}_{\mathbf{x}'_i}$ evaluated at $\mathbf{x}'_i$ that is generated by an inspection task where a gripper-held camera inspects an easily segmented blob positioned at $\mathbf{x}'_g$ (Hashimoto et al., 1992) where $\mathbf{J}^{-1}_{\mathbf{x}'_i}$ is defined by

$$\boldsymbol{\theta}_g - \boldsymbol{\theta}_i = \mathbf{J}^{-1}_{\mathbf{x}'_i}(\mathbf{x}'_g - \mathbf{x}'_i). \tag{5.16}$$

The calibrated feed-forward network approximation generates a correction delta

$$\Delta\boldsymbol{\theta}_{i+1} = \mathcal{F}(\mathbf{x}'_g - \mathbf{x}'_i, \boldsymbol{\theta}_i) \tag{5.17}$$

such that $\boldsymbol{\theta}_g \approx \boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_{i+1}$. To correct feed-forward network approximation error over the scale of measurement of $\mathbf{x}'_i - \mathbf{x}'_g$, (Hashimoto et al., 1992) used one feed-forward network to approximate gross $\mathcal{F}_g$ and one feed-forward network to approximate fine $\mathcal{F}_f$ positioning inverse task Jacobians. Repositioning of a manipulator to target position $\mathbf{x}'_g$ pixel measurement accuracy requires a composition of approximations, $\mathcal{F}_f$ after $\mathcal{F}_g$, i.e. two servo movements.

(van der Smagt, 1994) proposed a feed-forward network approximation F of a kinodynamic task field to servo a manipulator above an easily segmentable blob. The F kinodynamic approximation generates a joint acceleration correction delta

$$\Delta\ddot{\theta}_{i+1} = F(\mathbf{x}'_i - \mathbf{x}'_g, \theta_i, \dot{\theta}_i). \tag{5.18}$$

The forces acting on the manipulator $\tau_{i+1} = f(\Delta\ddot{\theta}_{i+1}, \dot{\theta}_i, \theta_i)$. See (1.4). The (van der Smagt, 1994) nonparametric approximation computes a servo sequence $\Delta\ddot{\theta}_{i+1}, i = 1, \ldots, n$. The final $\Delta\tau_{n+1}$ is constrained to decelerate the manipulator to bring it to an exact halt above the blob. F thus approximates a 'time-to-contact' kinodynamic solution approximation. The remainder of this Chapter analyses other interesting proposals for nonparametric approximations of manipulator kinematic functions in order to ground a fundamental analysis in §5.4 and §5.4 of the role of nonparametric regression in robotics and automation.

5.3.1 Table Lookup of Static and Differential Joint Free Space for Path Planning

(Mel, 1990) defines a $\sum\prod$ lookup table and an algorithm that support single camera image based planar path planning with a redundant 3-**R** manipulator. The camera segments images $\mathbf{x}'_i$ of white blobs on the shoulder, elbow and wrist joints of a 3-**R** Rhino manipulator and a target reaching goal $\mathbf{x}'_g$ and obstacles marked by white crosses. An incremental lookup table function $\sum\prod$ approximates:

1. a forward kinematic function F from joint space to thresholded joint image space $\mathbf{x}'_i$,
2. a visual task Jacobian field $\mathbf{J}$ defined with respect to the goal target $\mathbf{x}'_g$ and the current manipulator position in image space $\mathbf{x}'_i$. The approximation is similar to a path planning potential field solution. See (Latombe, 1991).

$\sum\prod$ maps the image vectors of the arm and the image velocity vectors with respect to the reaching goal crosses $\{\mathbf{x}'_i, \dot{\mathbf{x}}'_i\}$ to joint angles and joint velocities to $\{\theta_i, \dot{\theta}_i\}$. $(F, \mathbf{J})$ are approximated by random sampling Manipulator collisions occur when any part of the Rhino obscures an obstacle. Collisions define holes in the $\sum\prod$ approximation of static and differential Free Space $\sum\prod \approx (F, \mathbf{J})$. Once $\sum\prod$ has been approximated, a visual path planning function is defined. Path planning minimises $\mathbf{x}'_g, \mathbf{x}'_i$ by random forward search through $(F, \mathbf{J})$. If a path intersects a hole in $\sum\prod$, i.e. an obstacle in Free Space, the collision path is stored, a backtrack is initiated and a new random path is generated based on the contents of the list of collision paths. (Latombe, 1991) concludes a comprehensive review of robot path and motion planning theory with the observation that

> Massively parallel Connectionist architectures, e.g. Mel, may well push the field from using algebraic representations to distributed representations such as grids. ((Latombe, 1991), p.g. 588)

for instance (Barraquand and Latombe, 1991). To estimate $\sum\prod$ scaling complexity, assume a stereo-camera measurement of each manipulator joint position θ_i generates 4 measurements. Then $\sum\prod = (F, \mathbf{J})$ mapping involves table lookup approximation of two $\mathbb{R}^{28} \rightarrow \mathbb{R}^7$ functions. Assuming a camera discretisation range of $\bar{b}$ samples, the total sampling requirements for obstacle free $\sum\prod$ would be approximately $O(2 \times 7\bar{b}^{28})$. The inclusion of obstacles to generate a $\sum\prod$ Free Space would reduce sampling requirements depending on obstacle density, for instance to $O(2 \times 7(\frac{\bar{b}}{2})^{28})$.

A simpler path planner would be more efficient in terms of setup cost, compactness of Free Space representation and time required to compute a path to a target for the specific $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ path planning task that (Mel, 1990) considers. It is easy to observe that:

1. $\mathbb{R}^2$ image space can be calibrated to $\mathbb{R}^2$ Cartesian space by a 2×2 affine transform (no focusing or depth of field modelling necessary) that models camera optical axis offsets, rotation, skewing, and shearing effects by at least 4 manipulator movement samples. See §3.1.

2. A direct inverse function solution for a planar 3-**R** manipulator is derived in Appendix E parameterised by one joint angle θ_3. Let $\mathcal{F}^{-1} = (f_1, f_2)$ where $\theta_2 = f_1(x, y, \theta_3), \theta_1 = f_2(x, y, \theta_2, \theta_3)$.

3. If the goal target and obstacles are identified in image space and bounded by $\mathbf{O}_i$ loosely for instance by a circle or tightly by a deformable model minimised by an energy function, then the equations of $\mathbf{O}_i$ can be mapped by $\mathbf{A}$ to $\mathbf{O}_i^*$ in Cartesian space.

4. (Latombe, 1991) describes a variety of efficient methods to convert $\mathbf{O}_i^*$ algebraic obstacle representations to Free Space representations using $\mathcal{F}^{-1}$ for redundant manipulators.

5. Once Free Space is generated for instance as a directed graph at some quantisation of joint space, an efficient algorithm such as A^* may be used to generate an optimal, for instance shortest, path through the Free Space graph, and satisfy variable on-line path planning criteria. See (Latombe, 1991).

The outline of a path planner described above is more more setup cost, storage and path planning complexity efficient than the $\sum \prod$ lookup table and algorithm. The $\sum \prod$ table lookup representation of static and differential joint Free Space does not scale to higher dimensional visual path planning for anthropomorphic arms unlike a combination of parametric stereo-camera calibration to Cartesian space and algebraic approaches to path planning because of the combined curse of sampling dimensionality and perhaps the bias-variance regression regression tradeoff.

Calibration of parametric stereo-cameras to a manipulator requires not more than 20-50 samples if averaging over sensor noise is used as demonstrated in Chapter 3. For algebraic path planning, (Schwartz and Sharir, 1983) prove that where n is the *dof* of a manipulator, m is the number of polynomial constraints describing obstacles in Free Space and k is the maximum degree of such polynomials, then a complete algorithm of complexity $O((2k)^{3n+1}m^{2^n})$ exists to find a path to a target position.

5.3.2 Topographic Regression of Stereo Image Features on to Joint Coordinates

(Ritter et al., 1992) developed a model of sensorimotor coordination that was later applied to kinematic control of manipulators. The (Ritter et al., 1992) algorithm is based on a topographic map popularised by (Kohonen, 1989). Given data set $\{\mathbf{x}_i\} \in \mathbb{R}^n; i = 1, \ldots, n$, a topographic map uses a data clustering algorithm to order similar $\mathbf{x}_i$ where order is defined by a distance metric $[] \ldots []_n$.

5.6 Definition (A Topographic Map)

The addition of an annealing neighbourhood function N to batch k-means converts it into a topographic map $\mathbb{T}$. See (Kohonen, 1989). $\mathbb{T}$ is an approximate unconditional estimator of a density distribution $\mathbf{x}_i$. If $\mathbf{w}_l$ are the quantisation vectors of a $\mathbb{T}$, and given a distance metric $||\mathbf{w}_l - \mathbf{w}_i||$, a topographic map algorithm updates $\mathbf{w}_l^t$ at time $t+1$ for $\mathbf{x}_i$ by

$$\mathbf{w}_l^{t+1} = \mathbf{w}_l^t + \epsilon(t) N_t(||\mathbf{w}_c - \mathbf{w}_l||)(\mathbf{x}_i - \mathbf{w}_l^t) \quad \forall l \in \mathbb{T} \tag{5.19}$$

where $\mathbf{w}_c$ is

$$||\mathbf{x}_i - \mathbf{w}_c|| \leq ||\mathbf{x}_i - \mathbf{w}_l|| \quad \forall l \in \mathbb{T} \tag{5.20}$$

and $\epsilon(t)$ is a monotonic in t decreasing step size. $N_t(||\mathbf{w}_c - \mathbf{w}_l||)$ is a topographic order preservation function of a metric space.

If $N_t(||\mathbf{w}_c - \mathbf{w}_l||)$, a function of an Euclidean distance measure, is iterated by (5.19), as $t \rightarrow 10,000+$, $\mathbb{T}$ may converge to an approximate Voronoi tessellation of input probability density $P(\mathbf{x}_i)$ which over-estimates low probability and under-estimates high probability density. In the one dimensional $\mathbb{T}$ case, the unconditional probability density estimate is $\propto P(\mathbf{x}_i)^{2/3}$ around $\mathbf{x}_i$ (Hertz et al., 1991).

For greater than one dimensional topographic orderings, the topographic map update equation (5.19) has a complex stochastic transition probability structure. (5.19) transition probabilities can be described by a Fokker-Planck equation (Ritter et al., 1992). The (Ritter et al., 1992) Fokker-Planck equation proves that if the iteration of the topographic ordering is approaching an asymptotically stable state and if step sizes ϵ are small and if neighbourhood functions N_t are localised and if the probability distribution of input vectors $\mathbf{x}_i$ is uniform and if parameter fluctuations $\Delta\mathbf{w}_l = \mathbf{w}_l^{t+1} - \mathbf{w}_l^t$ are near equilibrium, then the probability distribution of $\Delta\mathbf{w}_l$ fluctuations in (5.19) converge to stable equilibrium, $\Delta\mathbf{w}_l \rightarrow 0$.

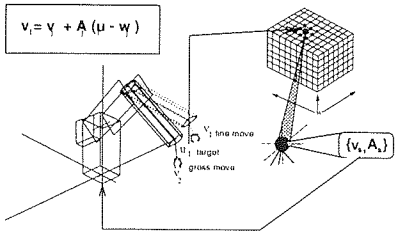

Figure 5.1: Topographic Regression of Image Features onto Joint Coordinates.

If the topographic map is far from convergence and other of the above conditions are not met, complex topographic disorder local minima structure may arise during (5.19) iteration. These minima structure appear as kinks in the topographic map when it, or slices of it if it is greater than three dimensions, are plotted. Local minima structure of the (5.19) iteration system are a function of initialisation of quantisation functions and hyper-parameters such as step size $e(t)$ and neighbourhood function N_t annealing schedules.

(Ritter et al., 1992; Walter and Schulten, 1993) extend the $\mathbb{T}$ model of unconditional density estimation to regression by definition of a multi-dimensional first-order Taylor expansion around $\mathbf{w}_l$, $\mathbb{S} = \{\mathbf{w}_l, \boldsymbol{\theta}_l, \mathbf{J}_l^{-1}\}$. The (Ritter et al., 1992; Walter and Schulten, 1993) topographic regression algorithm is closely liked to a regression of image measurements of a gripper position and location to joint vectors. The (Ritter et al., 1992; Walter and Schulten, 1993) algorithmic is tested with stereo-camera imaging of a manipulator holding a LED so this is described. In the (Ritter et al., 1992) experiments stereo-cameras output LED position $\{_1\mathbf{x}'_t, _2\mathbf{x}'_t\} = \mathbf{x}'_t \in \mathbb{S} \subset \mathbb{R}^4$. Since the (Ritter et al., 1992) algorithm is time dependent $\mathbb{S}^t = \{\mathbf{w}_l^t, \boldsymbol{\theta}_l^t, {}^t\mathbf{J}_l^{-1}\}$ vary with time. Each of the $\mathbf{w}_l^t$ of $\mathbb{T}$ defines a Voronoi cell $\mathbb{V}_l^t \subset \mathbb{S}^t$. $\{\mathbf{w}_l^t\}$ partition $\mathbb{S}^t$ into a Voronoi Tessellation. Each $\mathbf{w}_l^t$ has an associated output function ${}^tF_l^{-1}(\mathbf{x}'_t)$ to compute $\boldsymbol{\theta}_t^l$, the joint angles corresponding to the camera inputs $\mathbf{x}'_t \in \mathbb{V}_l^t$ at time t, i.e. the inverse kinematics

$$\boldsymbol{\theta}_t^l = {}^tF_l^{-1}(\mathbf{x}'_t) = \boldsymbol{\theta}_l^t + {}^t\mathbf{J}_l^{-1}(\mathbf{x}'_t - \mathbf{w}_l^t). \tag{5.21}$$

Each $\mathbf{w}_l^t$ is pointwise mapped to output $\boldsymbol{\theta}_l^t$. ${}^t\mathbf{J}_l^{-1}$ is a first order approximation at time t of the mapping around $\mathbf{w}_l^t$ defined by $\boldsymbol{\theta}_l^t - \boldsymbol{\theta}_{\mathbf{x}'_t}^l = {}^t\mathbf{J}_l^{-1}(\mathbf{w}_l^t - \mathbf{x}'_t)$. The parameters of $\{\mathbf{w}_l^t, \boldsymbol{\theta}_l^t, {}^t\mathbf{J}_l^{-1}\}$ are determined by the following algorithm:

1. At time t a random goal point $\boldsymbol{\theta}_t$ is generated and the manipulator is moved to this position.
2. Stereo-cameras observe corresponding input $\mathbf{x}'_t$.
3. $\mathbf{w}_c^t$ is determined such that $c = \underset{l}{\text{argmin}}(||\mathbf{x}'_t - \mathbf{w}_l^t||)$.

4. At $t+1$, the manipulator makes a gross positioning move to

$$\theta^{t+1} = \theta_c^t +^t \mathbf{J}_c^{-1}(\mathbf{x}'_t - \mathbf{w}_c^t). \tag{5.22}$$

5. image coordinates $\mathbf{x}'_{t+1}$ corresponding to θ^{t+1} are determined

6. At $t+2$, the manipulator makes a fine positioning correction move to

$$\theta^{t+2} = \theta_c^{t+1} +^t \mathbf{J}_c^{-1}(\mathbf{x}'_t - \mathbf{x}'_{t+1}). \tag{5.23}$$

7. image coordinates $\mathbf{x}'_{t+2}$ corresponding to θ^{t+2} are determined.

8. Correction terms are computed

$$\theta^{\Delta} = \theta_c^t +^t \mathbf{J}_c^{-1}(\mathbf{x}'_t - \mathbf{x}'_{t+1}), \tag{5.24}$$

$$\mathbf{J}^{-1^{\Delta}} =^t \mathbf{J}_c^{-1} +^t \mathbf{J}_c^{-1}(\mathbf{x}'_t - \mathbf{x}'_{t+2})\frac{(\mathbf{x}'_{t+2} - \mathbf{x}'_{t+1})^T}{||\mathbf{x}'_{t+2} - \mathbf{x}'_{t+1}||^2}. \tag{5.25}$$

9. Lattice element c approximation parameters and neighbours at distance q are updated using a Widrow-Hoff rule

$$\mathbf{w}_q^{t+2} = \mathbf{w}_q^t + \epsilon h_{qc}(\mathbf{x}'_t - \mathbf{w}_q^t), \tag{5.26}$$

$$(\theta, \mathbf{J}^{-1})_q^{t+2} = (\theta, \mathbf{J}^{-1})_q^t + \epsilon' h'_{qc}(\theta, \mathbf{J}^{-1})^{\Delta} - (\theta, \mathbf{J}^{-1})_q^t. \tag{5.27}$$

10. The neighbourhood functions h_{qc}, h'_{qc} may be Gaussian

$$h'_{qc}, h_{qc} = \exp\left(\frac{-||q-c||^2}{2\sigma^2(t)}\right), \tag{5.28}$$

or a rank distance function $q(r)$ of each lattice element l to $\mathbf{w}_c$

$$g(r) = \exp\left(\frac{-r}{z}\right) \tag{5.29}$$

where the rank[8] r of a lattice element l is an integer function of the distance of its reference vector $\mathbf{w}_l^r$ from $\mathbf{x}'_t$ and z plays a similar role to σ^2 in the Gaussian neighbourhood in $\mathbb{T}$.

Figure 5.1 is a representation of the (Ritter et al., 1992) algorithm. Iteration of the topographic network calibration cycle of (5.22) to (5.27) may cause $\mathbb{S}$ to converge to an approximation F of the function from stereo-image measurements of the manipulator feature to a single inverse branch. Restriction of the algorithm to movement of the manipulator to a single branch solution set is an algorithmic restriction since a multi-branch set of solutions will cause the algorithm to fail to converge since regression is ill-defined. If the F approximation converges, residual approximation error is non uniform. The regression accuracy of a piecewise linear mapping decreases as a function of distance of $\mathbf{x}_t$ from $\mathbf{w}_c^t$. Residual approximation error is highest for $\mathbf{x}_t$ on the boundary between different Voronoi cells $\mathbb{V}_l$. In order to smooth output θ_t as $\mathbf{x}_t$ crosses from V_l to V_k, a regression post-processor is defined by (Walter and Schulten, 1993) which averages outputs of local linear approximations based on the rank distance function

$$\theta_t = \frac{\sum_r g(r)(\theta_l^r +^r \mathbf{J}_l^{-1}(\mathbf{x}_t - \mathbf{w}_l^r))}{\sum_r g(r)}. \tag{5.31}$$

[8] A rank sequence $(0, 1, \ldots, n-1)$ is

$$||\mathbf{w}_l^0 - \mathbf{x}'_t|| < ||\mathbf{w}_l^1 - \mathbf{x}'_t|| < \ldots ||\mathbf{w}_l^{n-1} - \mathbf{x}'_t||. \tag{5.30}$$

(Walter and Schulten, 1993) benchmark the Gaussian neighbourhood function h_{qc}, h'_{qc} with the rank distance function and conclude that the rank neighbourhood function outperforms the Gaussian neighbourhood function. Once the Taylor expansion approximation has stabalised, for higher precision approximation (Ritter et al., 1992) employ multiple servo iterations through the Taylor expansion approximation of the mapping from stereo-camera space to joint space. With the rank neighbourhood function and fine position error correction servoing, (Walter and Schulten, 1993) report average error positioning error of 1.0% of the workspace range for approximation of a $\mathbb{R}^4 \rightarrow \mathbb{R}^3$ mapping generated by a real PUMA 560 observed by stereo-cameras. This precision, the highest reported in the literature on nonparametric regression of sensor measurements to joint angle functions, causes (Walter and Schulten, 1993) to

> conclude that the ... algorithm with collective neighbourhood learning provides an efficient, robust and accurate learning scheme for real world robot control. (p.g., 94.)

(Ritter et al., 1992) also implement a hierarchical organisation of the extended topographic map for regression of camera measurements onto a task redundant 5 *dof* manipulator: $\mathbb{R}^8 \rightarrow \mathbb{R}^5$.

In order to test and perhaps extend the (Ritter et al., 1992) algorithm, (5.19) to (5.31) were implemented in C on IRIX/RS3000 and in parallel C on transputers[9] for calibration of hierarchical extended topographic maps, i.e. 3 dimensional positioning extended topographic map with 3 dimensional extended orientation topographic maps attached to each node of the positioning extended topographic maps. See (Ritter et al., 1992). The (Ritter et al., 1992) algorithm was tested on calibration of stereo-cameras to an anthropomorphic arm simulated in a distributed robotics environment developed by (van der Smagt, 1994). Several hundred numerical experiments with both Gaussian and metric ranking neighbourhood functions verified the (Ritter et al., 1992) results to a reasonable accuracy (Doherty, 1993). Since (Ritter et al., 1992) do not specify an experimental design to determine values of hyper-parameters $\{p\} \equiv \{\epsilon_i, \epsilon_f, \delta_i, \delta_f, \epsilon'_i, \epsilon'_f, \sigma_i, \sigma_f, \sigma'_i, \sigma'_f, \rho_i, \rho_f, \rho'_i, \rho'_f\}$ for which the (Ritter et al., 1992) algorithm converges for the 6 *dof* manipulator mapping approximation, heuristic recursive golden section of hyper-parameter coordinates was used to determine a set of parameters that converge for a given extended topographic map controlling a given manipulator in a given workspace over several hundred experiments.

Several hundred partial and complete numerical experiments were required to debug the algorithm implemented on a serial processor and a MIMD transputer board. Plots of several hundred $\mathbb{S}$ algorithm convergence graphs made it clear that the algorithm convergence sensitivity to hyper-parameter values made the algorithm unsuitable[10] for robot control in a TQM ISO9000+ JIT CIM system. In addition (Ritter et al., 1992) contains no explicit projection of camera measurements to Cartesian space in order to integrate the proposed robot controller into a real world application in a CIM system. Even for the case of purely behavioural inverse kinematics, for stereo-camera based inverse kinematics of PUMA 560 the (Ritter et al., 1992) approximation:

1. is defined on only one inverse kinematic solution branch,

[9] Task farming parallelisation reduced the speed of the algorithm. The algorithm cannot be parallelised in time t. Spatial lattice parallelisation by loading topographic lattice subsets on to T805s to compute $\mathbf{x}_i - \mathbf{w}_l$ and parameter updates (5.26), (5.27) after $\boldsymbol{\theta}^{\Delta}$ and $\mathbf{J}^{-1\,\Delta}$ have been computed on a master processor requires slave processors to compute one vector subtraction and three vector additions per calibration cycle with the master processor doing most of the work. Inter-transputer communication delays during master processor propagation of $\mathbf{x}_i$ to slave processors, master processor collection of $\mathbf{x}_i - \mathbf{w}_l$ from the slave processors, and master processor propagation of (5.26), (5.27) to the slave processors made each calibration cycle slower on a 32 T805 board than a RS3000 SGI.

[10] The lattice Taylor approximation of the inverse kinematic solution element of a dextrous manipulator developed in Chapter 8 was however partly influenced by the replication and analysis of (Ritter et al., 1992; Walter and Schulten, 1993).

2. requires of the order of 10,000 calibration samples,

3. has many hyper-parameters which lead to a large parameter space to explore,

4. is based on locally linear maps $\{\mathbf{J}_l^{-1}\}$ that are not well defined near $\mathbb{CPS}$. Neighbourhood approximation kernels $h_{qc}, g(r)$ distort other kernels $\{\phi\} = \{\mathbf{J}_q^{-1}, \theta_q\}$.

Calibration of parametric models in §3 in contrast has no regression hyper-parameters except in the most advanced case of Levenberg Marquhardt regularisation and requires far fewer data samples. An analytical inverse solution exists for the PUMA 560 so sequential parametric model calibration results in calibration of image space to all 8 inverse solutions. (Ritter et al., 1992) argue that an advantage of their algorithm over sequential calibration of parametric models is that no a priori knowledge of a manipulator or cameras is required to develop an approximation of the mapping from image space to joint space. In particular, (Walter and Schulten, 1993) argue that their algorithm has an advantage over sequential calibration of parametric models when a camera has auto-focus. It is true that addition of a variable f parameter into classical sequential calibration of parametric cameras and DH parameters described Chapter 3 does marginally increase the complexity of the sequential calibration of parametric models. See (4.19).

5.3.3 Regularisation of a Dextrous Manipulator Inverse Kinematic Approximation

(Ritter et al., 1992) also implement a hierarchical organisation of the extended topographic map for regression of camera measurements onto a task redundant 5 *dof* manipulator: $\mathbb{R}^8 \rightarrow \mathbb{R}^5$. To resolve approximation redundancy a Euclidean neighbourhood distance function is constructed that makes $\mathbf{J}^{-1}$ converge to $\mathbf{J}^*$, a pseudoinverse. The solution thus eliminates a redundant *dof* from the solution manifold.

(Jordan and Rumelhart, 1992) use a combination of indirect inversion (vid. §2.5.4) and Lyapunov regularisation (vid. §2.6.3) to approximate dextrous manipulator inverse kinematics. The approximation is geared towards trajectory inversion. Different objective function regularisers, for instance $\frac{\partial^2 \mathbf{x}_i}{\partial \theta_j'^2} = c$, a smoothness regulariser, bias which particular inverse function is fitted (Jordan and Rumelhart, 1992). There is no explicit parameterisation of redundancy for run-time evaluation of solutions from different branch-manifolds in (Jordan and Rumelhart, 1992). (Jordan and Rumelhart, 1992) solutions are regularised statistical averages of the training data set and like (Ritter et al., 1992), a solution *dof* is discarded.

(Barhen et al., 1989) modify a Hopfield network (see (Hertz et al., 1991)) to approximate planar 3-**R** inverse kinematic mapping. Solutions are represented as sets of Hopfield network relaxation attractors. A Regularisation of a Hopfield energy function implements a parameterisation of redundant solutions so that a subset of redundant solutions may be available at run time. The computational complexity of the Hopfield network relaxation convergence to an approximate solution of the inverse kinematics of an industrial dimensional manipulator prohibit industrial deployment of the algorithm.

5.3.4 Approximation of a Canonical Coordinatisation of a Kinematic Solution Fibre

The inverse kinematic solution set of planar 3-**R** is homotopic[11] to a torus of solutions embedded in $\theta_1, \theta_2, \theta_3$ for each point $\mathbf{x}_i = (x_i, y_i)$ in a workspace $\mathfrak{W}$. The complete solution torus is a bundle

[11] See §8.2.1 for definition of differential geometric terms such as homotopy and fibre bundle.

of trivial fibre solutions over sub-workspaces $\mathfrak{W}_w$ separated by $\mathbb{CPS}$ of $\mathcal{F}$. See Definition 7.10. The topological structure of $f^{-1}(\mathbf{x}_i)$ can only change at $\mathbb{CPS}$, hence $f^{-1}(\mathbf{x}_i), \mathbf{x}_i \in \mathfrak{W}_w$ is diffeomorphic to $f^{-1}(\mathbf{x}_j), \mathbf{x}_j \in \mathfrak{W}_w$. The solution fibre $\mathfrak{F}$ is possibly in two disjoint pieces corresponding to two solution branches. See Definition 7.14. Each disjoint piece may be non-contractable along an embedding space dimension. See §8.2.2. (DeMers, 1993) developed a topographic map approximation to local fibre bundles for $\mathfrak{W}_w$-sheets between $\mathbb{CPS}$ of planar 3-$\mathbf{R}$ using the following algorithm:

1. generation of kinematic data $\{\boldsymbol{\theta}_i, \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)\}$,
2. storage of all $\mathbf{x}_i \in \mathfrak{W}$ within a given subspaces of $\mathbb{R}^2$ in a K-D tree (i.e. a K-2 tree for planar Cartesian space), with associated storage of all $\boldsymbol{\theta}_i$ associated with each $\mathbf{x}_i$.
3. $\forall \mathbf{x}_i \in$ a given K-D subspace, k-means clustering of $\boldsymbol{\theta}_i$ associated with all $\mathbf{x}_i$ in the subspace determines b, the number of inverse solution branches of $\mathcal{F}$ in that subspace.
4. $\forall \mathbf{x}_i, \forall \boldsymbol{\theta}_i$ are assigned a solution branch[12] label $1 < a < b$ ($b = 1, 2$ in the case of 3-$\mathbf{R}$) by comparison with neighbours under an assumption of continuity of $\mathcal{F}$.
5. if a changes between K-D subspaces, this indicates the presence of a $\mathbb{CPS}$ between $\mathfrak{W}_w$ and $\mathfrak{W}_{w+1}$ where w indexes $\mathbb{CPS}$ separated partitions of the base space. This can be confirmed by determination if $|\mathbf{J}| \approx 0$ in this vicinity.
6. After $\mathbb{CPS}$ are identified, $(\mathbf{x}_i, \boldsymbol{\theta}_i) \in \mathbb{CPS}$ are removed from the K-2 tree,
7. A canonical parameterisation[13] of the two well defined local fibre bundle branch $\mathfrak{F}_1, \mathfrak{F}_2$ structures attached to the base space $\mathfrak{W}_w$-sheets can then be defined after the homotopy class of each fibre is determined[14].
8. (DeMers, 1993) analytically enumerates the $\mathfrak{F}_1, \mathfrak{F}_2$ homotopies to T, either contractable or disjoint, for each $\mathfrak{W}_w$-sheet.
9. $\forall \mathbf{x}_i$, a given K-D subspace of each $\mathbb{CPS}$-free $\mathfrak{W}_w$-sheet, (DeMers and Kreutz-Delgado, 1991) fit either one (contractable fibre) or two (disjoint fibre) topographic maps of topography T to all of the $\boldsymbol{\theta}_i$ in the given K-D subspace depending of the fibre homotopy identified analytically in the base space. A canonical coordinatisation ζ_w of the T homotopic solution fibres in each $\mathfrak{W}_w$ is defined by fixing a corresponding origin ζ_w in each topographic map coordinate system which preserves continuity of coordinatisation between neighbouring fitted T fibres of all K-D subspaces in a given $\mathfrak{W}_w$-sheet.

The set of canonically ζ_w coordinatised topographic maps in $\mathfrak{W}_w$ implements a discrete field approximation Γ_w of solution branch fibres over $\mathfrak{W}_w$. In order to compute inverse kinematic solutions for $\mathbf{x}_m$ that are not part of the calibration set $(\boldsymbol{\theta}_i, \mathbf{x}_i)$, nearest neighbour interpolation is used between neighbouring topographic map elements to define a piecewise linear canonical coordinatisation Γ_w where $\boldsymbol{\theta}_m = \Gamma_w(\mathbf{x}_m \in \mathfrak{W}_w, \zeta_w, a)$.

[12] A branch is a contractable or a non-contractable disjoint solution fibre. See Definition 7.11.

[13] (DeMers, 1993) also applies the same procedure to the degenerate case of approximation of inverse kinematics of planar 2-$\mathbf{R}$, where b direct inverse functions $\boldsymbol{\theta}_b = f_b(\mathbf{x}_b)$ are approximated by separate feed-forward networks after branch identification.

[14] (DeMers, 1993) discusses estimation of fibre homotopy structure using S^1 topographic mappings. (DeMers, 1993) proposes a heuristic to determine the homotopy class of a maximally singularity free connected sub-workspace based on a canonical coordinatisation of a single $\mathbf{x}_i$ query point. If a S^1 topographic map can be fitted to the data, then this indicates that the fibre in this region has non-disjoint structure. If the data cannot be well fitted by a S^1 topographic mapping, this indicates disjoint fibre structure.

Γ_w, Γ_{w+1} for each $\mathfrak{W}_w, \mathfrak{W}_{w+1}$ neighbour combination are not guaranteed to vary smoothly as a function of ζ_w, ζ_{w+1} between $\mathfrak{W}_w$-sheets and in general they will not and a canonical parameterisation discontinuity may exist between ζ_w, ζ_{w+1} crossing $\mathbb{CPS}$ between $\mathfrak{W}_w$ and $\mathfrak{W}_{w+1}$. For $\mathbf{x}_n \in \mathbb{C}$ between $\mathfrak{W}_w$ and $\mathfrak{W}_{w+1}$ such that

$$\Gamma_w(\mathbf{x}_m, \zeta_w, a) \approx \Gamma_{w+1}(\mathbf{x}_m, \zeta_{w+1}, a). \tag{5.32}$$

In general $\zeta_w \neq \zeta_{w+1}$ if there are more than two $\mathfrak{W}_i$-sheets. If $\zeta_w = \zeta_{w+1}$ on either side of a $\mathbb{CPS}$, then the origin of canonical coordinatisations of solution fibres of $\mathfrak{W}_w$ and $\mathfrak{W}_{w+1}$ must be close to the $\mathbb{CPS}$. If there are more than 2 $\mathfrak{W}_w$-sheets, then a canonical coordinatisation discontinuity between ζ_{w+1}, ζ_{j+2} must occur across $\Gamma_{w+1}, \Gamma_{w+1}$.

(DeMers, 1993)'s topographic map canonical coordinatisation of the solution fibre bundle of planar 3-**R** is elegant. However the canonical parameterisation does not easily scale to approximation of a 7 *dof* homomorphism because of the curse of sampling dimensionality.

Chapter 8 defines a local linear approximation of a direct inverse function coordinatisation of the solution element of the solution group of the fibre bundle of planar 3-**R**. The local linear approximation scales to approximation of the solution element fibre bundle of a dextrous manipulator, the RR-K1207. The Chapter 8 local linear approximation of a redundant inverse mapping can parameterise the complete redundant manifold of inverse solutions efficiently. The approximation developed in Chapter 8 was influenced by the (Ritter et al., 1992) extended topographic map regression algorithm and the topographic map approximation of a canonical coordinatisation of an inverse kinematic solution fibre of a dextrous manipulator developed by (DeMers, 1993).

5.4 The Role of Nonparametric Regression in Robotics and Automation

The goal of the analysis of this chapter was to evaluate the hypothesis proposed in §1.2 on pg. 30 that nonparametric regression components have applications proposed manipulator control systems. The statistical levels at which nonparametric regression problems may be solved range from maximum likelihood regression to Bayesian optimal nonparametric model selection and optimal nonparametric regression. A potential advantage of nonparametric regression over sequential calibration of parametric models of cameras and manipulator kinematics is that if no a priori parametric modelling is required, this can potentially simplify solutions and thus considerably cheapen solution costs. Nonparametric regression can potentially meet an essential engineering heuristic - KISS (keep it (a) simple solution). Parametric analysis is in general expensive in engineering labour costs and time. If an engineer can apply a simple black box nonparametric regression procedure to solve a problem, the solution may be cheaper and quicker to generate than that generated by a parametric analysis.

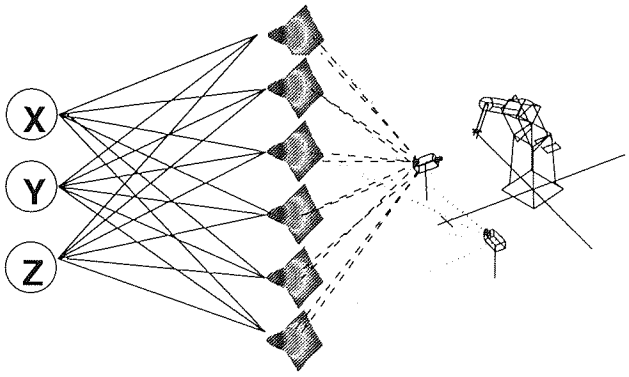

Figure 5.2: Radial Basis Function Regression of Stereo-image Features onto Cartesian Space.

This is illustrated with a simple numerical experiment in which radial basis functions approximate a $\mathbb{R}^4 \rightarrow \mathbb{R}^3$ mapping from simulated stereo-cameras onto PUMA 560 joint space. The data is generated

by $\mathcal{F}$ of the PUMA 560 in Table 2.1 and the camera model from §4.3.2 with zero lens distortion, focal length = 35mm, $X = 0m, Y = 2m; Z = .9m$ offset of image centre from Cartesian space, and angle of rotation $\theta = 0^o$, and elevation $\alpha = -\pi/2$. $6 \times 6 \times 6 = 216$ calibration data samples are generated $\boldsymbol{\theta}_i : \theta_1, \theta_2, \theta_3 = 0.0, 0.2, 0.4, \ldots, 1.0$ radian increments with $\theta_4, \theta_5, \theta_6 = 0.0$. A test set is generated with $\theta_1, \theta_2, \theta_3 = 0.1, 0.3, 0.5, 0.7, 0.9$ radians.

25 numerical experiments were performed to estimate the error of radial basis function regression of stereo-image features onto joint space as a function of the number of basis kernels. 25 numerical experiments were also performed to estimate the error of radial basis function regression of stereo-image features onto Cartesian space as a function of the number of basis kernels. See Figure 5.4. The number of kernels was initialised at 2 and incremented by 2 to 50 kernels in 25 experiments. Radial basis functions were initialised with centres $\mathbf{c}_k$ on a random subset of data, and spherical kernel variance of $\sigma^2 = 1.3$ input scale units was determined to give adequate regression smoothing after a few preliminary regression trials. Radial basis function coefficients were determined by Singular Value Decomposition (SVD). Figure 5.3 and Figure 5.4 show that on both regression problems the MSE measure decreases until about 50 out of 216 possible basis functions are reached. With more than 50 basis functions, SVD inversion of a singular basis function correlation matrix causes error to fluctuate. The basis function correlation matrix looses rank when a basis function is introduced which cannot differentiate between 2 or more input patterns. When rank loss occurs is a function of the basis function variances. If the variances of the basis functions are reduced as the number of basis functions is increased, then this increases the probability of full rank of the basis function correlation matrix. Figure 5.3 shows that for radial basis function approximation of the stereo-camera model, MSE minimum is approximately $10^{-10}m^2$. Figure 5.2 shows that the radial basis function approximation of the complete mapping from stereo-camera measurements to joint coordinates has a minimum MSE of 0.04 radians.

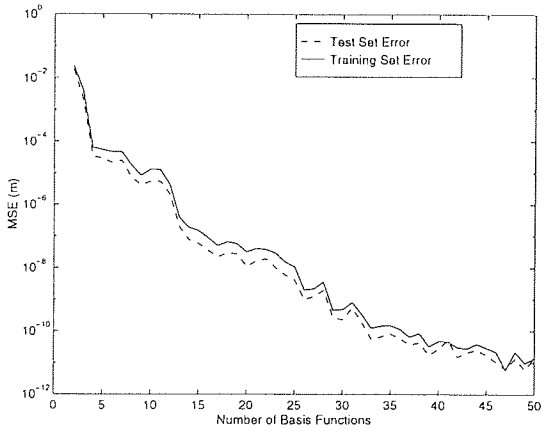

Figure 5.3: The MSE of Radial Basis Function Regression of Stereo-image Features onto Cartesian Space as a Function of the Number of Basis Kernels

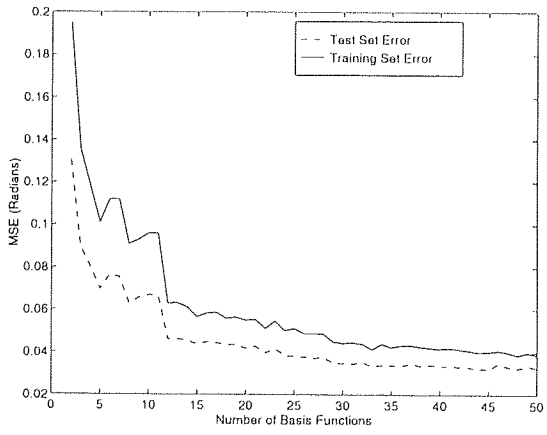

Figure 5.4: The MSE of Radial Basis Function Regression of Stereo-image Features onto Joint Space as a Function of the Number of Basis Kernels.

In order to increase precision on the calibration and test sets, sampling density could be increased and more complex kernel parameter estimation procedures could be employed such as cross-validation, nonlinear optimisation of elliptical variances or Bayesian regularisation of noise fitting. See (Bishop, 1995; Ripley, 1996). However unless these procedures are fully automatic, the necessity for a software developer to specify values for hyper-parameters reduces the potential KISS advantage in terms of

solution cost and generation time of a nonparametric regression over sequential calibration of parametric models. To implement this simple experiment, spherical kernel variance had to be determined by some initial hyper-parameter tweaking and a technician would be required to select the number of basis functions that generates a good solution approximation.

This chapter has analysed several examples of the application of complicated nonparametric regression methods to solve simple parametric regression problems, for instance the (Ritter et al., 1992) regression of stereo-images to joint angles. Often a simple analytical problem has been solved by complicated nonparametric regression. For instance, it requires more effort to understand, implement, and determine convergence hyper-parameters for the (Ritter et al., 1992) algorithm than to deploy sequential calibration of parametric models of cameras and manipulator kinematics. This is a key issue for industrial deployment of nonparametric regression models. A nonparametric regression procedure which for instance requires an engineer to be capable of Bayesian prior generation could be difficult to deploy in many industrial applications if engineers have difficulty generating a Bayesian regression prior.

Despite the many interesting nonparametric control systems proposed by for instance (Miller, 1989; Miller et al., 1990; Kuperstein and Rubinstein, 1989; Cooperstock and Milios, 1993; Mel, 1990; Ritter et al., 1992; Walter and Schulten, 1993; Jordan and Rumelhart, 1992; Barhen et al., 1989; Kuperstein and Rubinstein, 1989; DeMers, 1993; Kawato et al., 1987; van der Smagt, 1994; Edelman, 1989; Eckmiller, 1989; Hashimoto et al., 1992), there have been no applications of nonparametric regression deployed in industrial robotics and automation control systems despite the potential KISS advantage of nonparametric regression over parametric regression.

There are no commercial applications of nonparametric regression at kinematic control levels, including path planning, of robotics and automation systems for two basic reasons:

1. nonparametric solutions are easy to implement but difficult to verify and validate without recourse to advanced statistical methods. See (Bishop, 1995),

2. nonparametric solutions are difficult to scale.

The curse of dimensionality and the bias-variance tradeoff in nonparametric estimation are the major barriers to increasing functionalities of nonparametric robotics. These define the regression limits of nonparametric behaviourist robotics in addition to the general representational limits of behaviourist control defined by (Chomsky, 1954)'s review of (Skinner, 1953). Nonparametric methods reduce requirements for a priori analysis but increase data requirements and have non trivial extrapolation for a posteriori validation of solutions (Geman et al., 1992). Data collection constraints and the high dimensionality of many industrial nonparametric regression problems account for the relative paucity of industrial applications of behavioural manipulator control based on nonparametric models. In general, parametric analytical robotic control systems are more mathematically reliable and scalable than nonparametric ones.

At the robot inverse dynamic control level, nonparametric models have also been proposed for approximation of dynamic control components such as friction, stiction, Coriolis and centrifugal forces (Warwick et al., 1992; White and Sofge, 1992; Miller III et al., 1990; Miller, 1989; Miller et al., 1990). The absence of industrial deployment of nonparametric regression at kinematic (and to a lesser extent at dynamic levels) within TQM JIT hierarchical functional robotics and automation systems is mainly due to the fact that analytical models are comparatively relatively straightforward to derive, implement, verify and validate. The arguments by some behavioural control system developers that nonparametric models can be implemented more efficiently for real-time system operation than analytical models also do not survive scrutiny in the case of kinematic level applications. Closed-form

inverse kinematic solutions and Jacobian evaluations are compact and relatively efficient and can be parallelised to some extent and integrated on special purpose hardware (Lee, 1991a; Zhang and Paul, 1991).

The absence of commercial application depolyment of nonparametric regression at kinematic and to a lesser extent dynamic control levels of robotics and automation systems may not extend to higher control levels. Nonparametric models may perhaps be justifiably applied at higher levels of robotics and automation systems if analytical modelling fails, usually close to economic and production scheduling levels. Optimisation of global production efficiency-flexibility (see (1.1) and (1.2)) may require a nonparametric model to integrate various parametric production model outputs with an error function specified in a financial/accounting coordinate system. A good example of the utility of nonparametric models is provided by (Breiman et al., 1984) where a classification and regression tree is used to approximate an optimum production function where it is difficult to model the complete production function because of the different scales of the data. For instance, it is hard to combine data related to process operation, for instance temperature and pressure, with human factors data, for instance average attention span of operators as a function of time since shift initialisation and level of production team morale, in order to simulated advanced TQM JIT production system performance.

Nonparametric models may also be useful in simulation of distributed real-time robotics and automation systems. See §6.5.3. Nonparametric models may glue parametric analytical complexity measures of code blocks and Markov models of processing queues together to compute a global performance index. The nonparametric model must be easily differentiable, for instance a feed-forward network, in order to compute error gradients to optimise analytical complexity models and Markov parameters in order to determine the optimal operational profile of the system being designed.

Tactical KISS nonparametric solution deployment optimisation involves simultaneous minimisation of solution cost and time to market and maximisation of solution quality and durability and extendability and maintainability. It is un-competitive if an optimal parametric solution arrives too late at the market and is too expensive because an engineer spent too long deriving an analytical model. For non-critical applications, i.e. where there are no reliability, safety, availability, and robustness control constraints, economic circumstances may dictate that for instance a hierarchical cross validation nonparametric regression on a low dimensional data set if it can be quickly and adequately deployed by an engineer or software developer is an adequate solution given solution cost and delivery time constraints. The next section presents an automatic nonparametric regression procedure and illustrates its deployment on a regression problem.

5.5 Automatic Optimal Sequential Linear and Nonparametric Regression

This section illustrates a KISS nonparametric regression deployment heuristic - the optimal representational loading of nonparametric model parameters occurs when a model is fit to a pure nonlinear residual regression error remaining after a linear regression. This section also demonstrates a completely automatic procedure for determination of optimal nonparametric model complexity and parameter values for regression to an unbiased estimate of an underlying regression function. The automatic procedure is suitable for deployment by an engineer or software developer that is unfamiliar with nonparametric regression theory since the procedure has no regression hyper-parameters. The procedure is illustrated with an application where a nonparametric model corrects camera auto-focus, off-set, and depth of field residual error distortions of an affine stereo-camera model. The application:

1. validates the rapid numerical algorithm prototyping environment for distributed robotic system control defined and implemented in Chapter 6.

2. demonstrates a general completely automatic black box procedure that could be applied by technicians to low dimensional regression problems at higher levels of robotics and automation systems

5.5.1 Generation of Calibration Data Samples

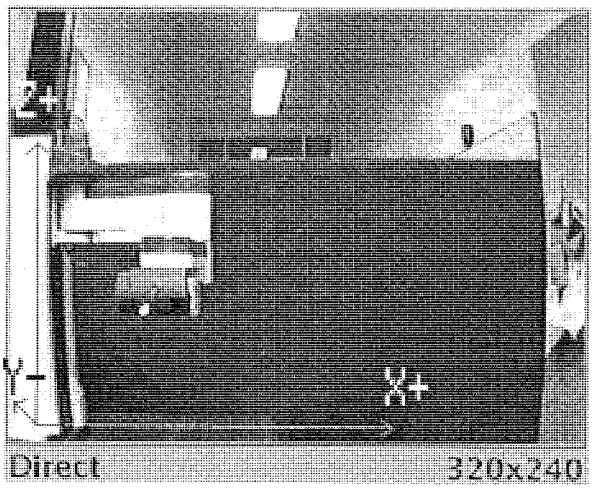

Figure 5.5: The Cartesian Coordinate System Viewed by Camera One.

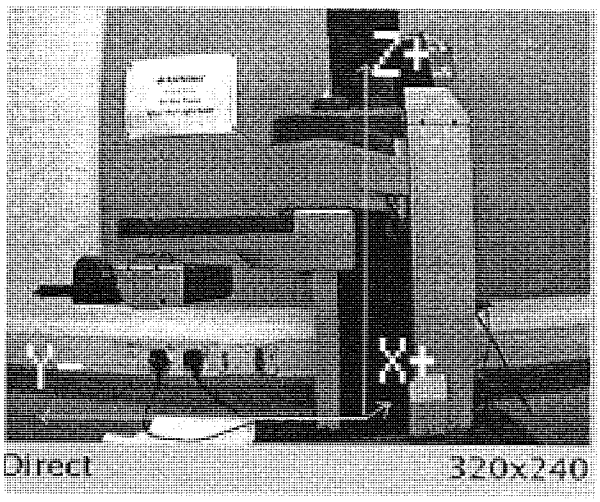

Figure 5.6: The Cartesian Coordinate System Viewed by Camera Two.

A right handed $\mathfrak{C}$ coordinate system is attached to the RT100 of Table 6.1.1. Figure 5.5 shows the right handed $\mathfrak{C}$ coordinates system from the point of view of camera 1 with the $Y-$, the negative Y axis, pointing out of the picture at the viewer. Camera 1 is placed approximately $1.5m$ along $Y-$. Figure 5.6 shows the right handed $\mathfrak{C}$ coordinates system from the point of view of camera 2 with the $X+$ pointing out of the picture at the viewer. Camera 2 is placed about $1.5m$ along the $X+$ axis.

The distance of the CCD from the XY plane is approximately 0.4m for each camera. The cameras are placed by hand so that their optical axes are approximately orthogonal to the CCD sensor plane facing the origin. One camera has zero yaw and the other camera has approximately $\pi/2$ yaw in a fixed angle XYZ orientation coordinate system. There is approximately zero camera roll and pitch in $\mathfrak{C}$ space.

The primary sampling constraint is to restrict the LED's location within the field of view of both cameras simultaneously. See Figure 5.5 and Figure 5.6. With the cameras placed as described above, and with wrist roll, pitch and yaw parameters all zero, i.e. the wrist points straight ahead along the axis of the forearm, a sweep of the arm from $600mm \leq \theta_1 \leq 800mm$, $-40^o \leq \theta_2 \leq 0^o$, $-40^o \leq \theta_3 \leq 20^o$ keeps the arm within both images while traversing most of the image space of both cameras.

Discretisation of the within-image ranges of each $\theta_1, \theta_2, \theta_3$ into 5 samples including the inequality end points generates 125 $\boldsymbol{\theta}_i$ data points. See Figure 5.7,Figure 5.8,Figure 5.9. Command of the RT100 to move to $\boldsymbol{\theta}_i, i = 1, \ldots 125$ followed by segmentation of the LED images from the 620x480 pixel images by the algorithm of §6.1.2 generates corresponding 125 LED stereo-image coordinates $\mathbf{x}'_i$. These LED image coordinates from the optical centre of the CCDs are presented in Figure 5.10, Figure 5.11,Figure 5.12,Figure 5.13.

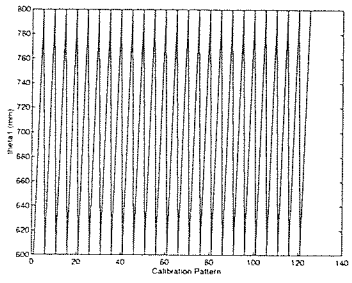

Figure 5.7: The Calibration Data for θ_1.

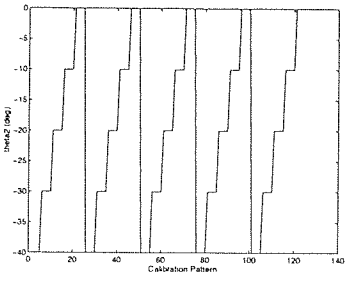

Figure 5.8: The Calibration Data for θ_2.

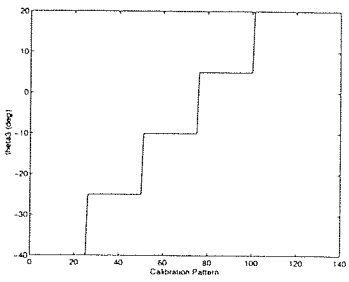

Figure 5.9: The Calibration Data for θ_3.

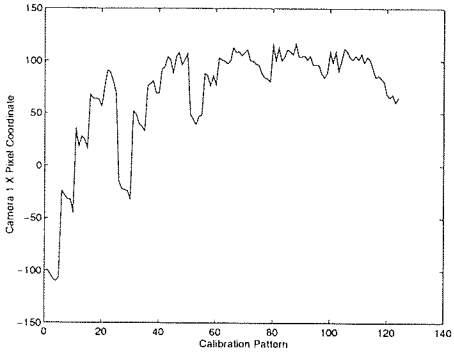

Figure 5.10: The Calibration x_i' Data for Camera One.

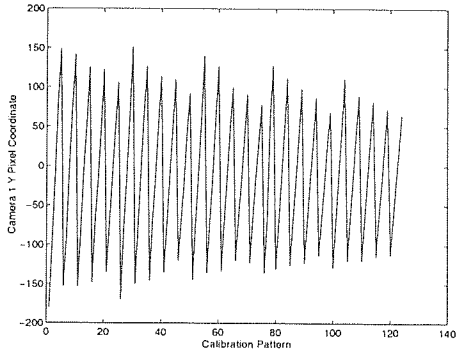

Figure 5.11: The Calibration y_i' Data for Camera One.

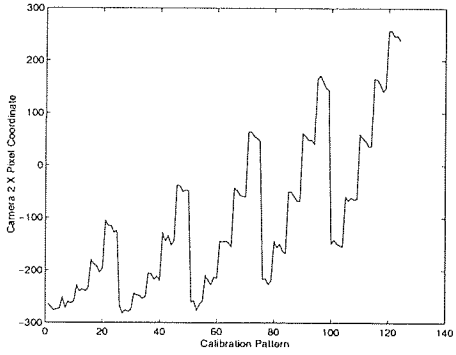

Figure 5.12: The Calibration x_i' Data for Camera Two.

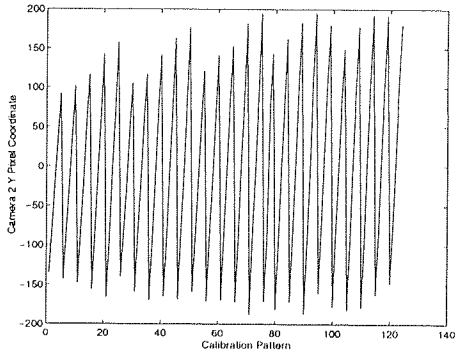

Figure 5.13: The Calibration y_i' Data for Camera Two.

5.5.2 Cross Validation of an Affine Stereo-camera Model

$\mathbf{S}$, an affine stereo model, projection of objects onto images is reasonably accurate for measurements which do not vary widely in the depth of field. In an affine stereo model, each $\mathbf{x}_j$ coordinate has x, y, z components. If stereo-image measurements are rearranged into homogeneous form, i.e. $\mathbf{x}_j = [x, y, z, 1]$, let 4 $\mathbf{x}_j$ points be organised into a 4×4 $\mathbf{X}$ matrix, i.e. each $\mathbf{x}_j$ is row j of $\mathbf{X}$. Let each $\mathbf{x}_j'$ image coordinates x_1, y_1 for camera 1 and x_2, y_2 for camera 2 be organised into a 4×4 matrix $\mathbf{X}'$ where each $\mathbf{x}_j'$ is column j of $\mathbf{X}'$. An affine model $\mathbf{S}$ of stereo is

$$\mathbf{S} : \mathbb{R}^4 \rightarrow \mathbb{R}^3 : \mathbf{X} = \mathbf{S}\mathbf{X}' \tag{5.33}$$

where

$$\mathbf{S} = \mathbf{X}\mathbf{X}'^{-1}. \tag{5.34}$$

$\mathbf{S}$ is a 4×4 matrix. $\mathbf{x}_i$, $i = 1, \ldots, 4$, non-coplanar $\mathfrak{C}$ samples are required to determine parameters of $\mathbf{S}$ (5.34). The parameters of $\mathbf{S}$ can be determined by selection of at least four maximally non-coplanar[15] samples $\mathbf{x}_i, i = 1, \ldots, 4$ from a calibration data set followed by computation of pseudoinverse fit of $\mathbf{S}$ to the data by (5.34). The error of the $\mathbf{S}$ approximation of the regression from image space to Cartesian space reduce precision of sensor based inverse kinematic control or subsequent re-calibration of kinematics. Calibration points $\mathbf{x}_j$ beyond the parallelogram defined by the $\mathbf{x}_j$ used to calibrate $\mathbf{S}$ may have significant difference in $\backslash z_j$, distance from the optical centre of camera CCD compared to average $\backslash \bar{z}$ of $\mathbf{x}_i$ used for calibration of $\mathbf{S}$ parameters. This generates affine stereo errors as a function of depth of field differences $\backslash z_j - \backslash \bar{z}$.

$\mathbf{S}$ determined by pseudoinversion of the four maximally non-coplanar calibration data points is not necessarily optimal if one of the points is an outlier for instance. The parameters of $\mathbf{S}$ are more robustly determined by k-fold cross validated Singular Valued Decomposition (SVD). Let the data set $\mathbb{D}$ be divided into $m = 5$ pieces $(V_1, \ldots, V_m)$. V_1 is composed of elements $1, \ldots, 25$ of $\mathbb{D}$, V_2 is composed of elements $26, \ldots, 50$, V_3 is composed of elements $51, \ldots, 75$, V_4 is composed of elements $76, \ldots, 100$ and V_5 is composed of elements $101, \ldots, 125$. Each V_i contains 25 elements. $V_{-i} = \mathbb{D} \backslash V_i$. Let $\mathbf{S}_i$ be determined by $\mathbf{X}_{-i}(\mathbf{X}'_{-i})^+$ where the pseudoinverse is determined by SVD. See (Press et al., 1992). The root mean square error (RMSE) of $\mathbf{S}_i$ on test set V_i is the sum of the squares of the elements of $\mathbf{S}_i\mathbf{X}'_i - \mathbf{X}_i$ divided by the number of rows, in this case 25. Similarly, the RMSE error of $\mathbf{S}_i$ on the calibration set V_{-i} is the sum of the squares of the elements of $\mathbf{X}_{-i} - \mathbf{S}_i\mathbf{X}'_{-i}$ divided by the number of rows, in this case 100. Table 5.3 displays the RMSE errors for $\mathbf{S}_i$ on V_i and V_{-i}.

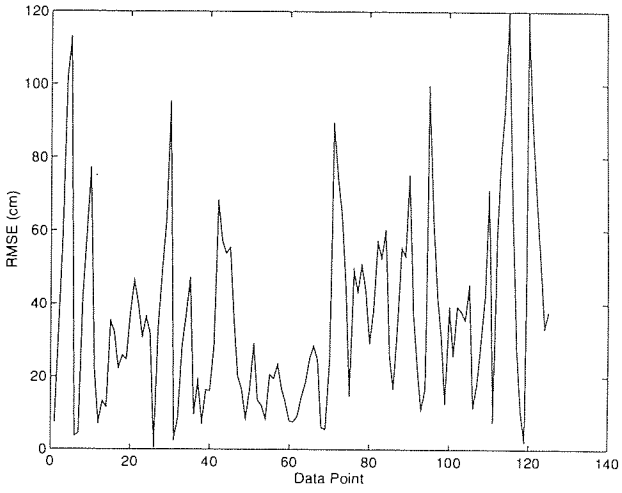

Figure 5.14: The RMSE of the Optimal Affine Stereo Model S_3 determined by Cross Validated SVD.

The maximum, mean and variance of the absolute error residuals of $\mathbf{S}_i$ are tabulated in Table 5.4, Table 5.5, Table 5.6, Table 5.7 and Table 5.8. Visual examination reveals a range of variation of the values in Table 5.4, Table 5.5, Table 5.6, Table 5.7 and Table 5.8. To determine the significance of this variation it is necessary[16] to perform a set of t-tests or an analysis of variance (ANOVA).

Examination of Table 5.3 reveals that $\mathbf{S}_3$ has the lowest MSE on test set V_3. Hence $\mathbf{S}_3$ is selected as the optimal cross validated affine stereo model $\mathbf{S}_*$. For the optimal cross-validated affine model $\mathbf{S}_3$, it can be inferred from Table 5.6 that the average absolute error over all coordinate is $39.95cm$ on the calibration set and $24.68cm$ on the test set. The RMSE of an $\mathbf{S}_3$ transformation summed over coordinates is presented for each element of the data set in Figure 5.14.

[15] If $\mathbf{x}_i$ are row stacked into a 4×4 matrix $\mathbf{X}$, a four level combinatoric loop over all $\mathbf{x}_i$ is required to determine $\mathbf{x}_i$ that max(rank($\mathbf{X}$)), i.e the four maximally non-coplanar $\mathbf{x}_i$.

[16] This is not undertaken here since while statistical analysis of linear regression is interesting, it serves little useful purpose at the level of generation of a robust affine stereo-model since statistical analysis is in general not automatic. The introduction of statistical analysis would remove the KISS advantage of the proposed hierarchical cross validation nonparametric regression over sequential parametric calibration.

$\mathbf{S}_i$	RMSE(V_{-i}) in m	RMSE(V_i) in m
$\mathbf{S}_1$	0.2434	0.5738
$\mathbf{S}_2$	0.2794	0.3388
$\mathbf{S}_3$	0.3002	0.1980
$\mathbf{S}_4$	0.2774	0.3196
$\mathbf{S}_5$	0.2532	0.4083

Table 5.3: The RMSE on calibration V_{-i} and test V_i sets of Affine Stereo Models determined by SVD.

	Calibration Error on V_{-1}			Test Error on V_1		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.6078	0.1932	0.5643	0.8834	0.2961	0.8147
mean	0.1272	0.0595	0.1234	0.3082	0.1487	0.2826
variance	0.0117	0.0016	0.0113	0.0719	0.0050	0.0609

Table 5.4: Absolute Error Statistics of the Affine Stereo Projection S_1 Determined by SVD.

	Calibration Error on V_{-2}			Test Error on V_2		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.5854	0.1782	0.5468	0.5523	0.1784	0.5151
mean	0.1419	0.0651	0.1404	0.2012	0.0941	0.1770
variance	0.0170	0.0024	0.0150	0.0173	0.0017	0.0165

Table 5.5: Absolute Error Statistics of the Affine Stereo Projection S_2 Determined by SVD.

	Calibration Error on V_{-3}			Test Error on V_3		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.5378	0.1742	0.5645	0.4123	0.1722	0.3116
mean	0.1653	0.0746	0.1596	0.1065	0.0525	0.0878
variance	0.0155	0.0019	0.0147	0.0098	0.0018	0.0064

Table 5.6: Absolute Error Statistics of the Affine Stereo Projection S_3 Determined by SVD.

	Calibration Error on V_{-4}			Test Error on V_4		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.5854	0.1784	0.5468	0.4231	0.1527	0.5048
mean	0.1598	0.0721	0.1496	0.1859	0.0902	0.1895
variance	0.0201	0.0027	0.0171	0.0099	0.0017	0.0129

Table 5.7: Absolute Error Statistics of the Affine Stereo Projection S_4 Determined by SVD.

	Calibration Error on V_{-5}			Test Error on V_5		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.4758	0.1598	0.5512	0.5859	0.1870	0.6607
mean	0.1381	0.0654	0.1284	0.2303	0.0932	0.2274
variance	0.0121	0.0018	0.0107	0.0266	0.0023	0.0266

Table 5.8: Absolute Error Statistics of the Affine Stereo Projection S_5 Determined by SVD.

5.5.3 Hierarchical Cross Validation of a Nonparametric Regression

$\mathbf{S}_*$ models translation, rotation, skewing, shearing, and scaling transformations of image space to Cartesian space. See §3.1. After $\mathbf{S}^*$ is determined, radial basis functions F can be calibrated to the data set in order to minimise

$$E = \frac{1}{2}\sum_j (F(\mathbf{S}^*\mathbf{x}'_j) - \mathbf{x}_j))^2. \tag{5.35}$$

A nonparametric model can be fit to the residual non linear regression errors defined in Table 5.3 and Table 5.5. The hierarchical k-fold cross validation procedure described in §5.2.1 is a completely automatic procedure to determine optimal nonparametric model complexity and parameter values for regression to an unbiased estimate of an underlying regression function. Such an estimate can correct calibration error residuals remaining after calibration of $\mathbf{S}$ an affine stereo model to Cartesian space that are caused by:

1. auto-focus,
2. depth of field,
3. radial lens distortion[17],
4. lens centre offsets,
5. non-orthogonality of optical axis to the centre of a CCD surface plane.

In this section, radial basis functions model these distortions of the affine stereo camera model. The bias and variance of a radial basis function estimator is a function of:

1. the number and centre position of basis kernels,
2. the variance $\boldsymbol{\sigma}_j^2$ of the kernel functions ϕ_j,
3. the underlying regression function that is being approximated,
4. the noise levels of the sample data. The noise in image measurements of the LED is a complicated function of the interaction of the 50hz AC fluroscent strip strobing, the natural illumination variation, and the quantization of the CCD cells as a function of LED distance from the image planes. See §6.1.2.1.

If the radial basis function model complexity is approximately correct, i.e. the number of basis functions is adequate to fit the data quite well, the regression variance can be controlled by setting the variance parameters $\boldsymbol{\sigma}_j^2$ of the kernel functions. Data can be fitted more closely with more localised ϕ_j. A small value of $\boldsymbol{\sigma}_j^2$ with respect to the mean separation between calibration data samples has a tendency to reduce bias and increase variance of radial basis function fit to a data set. If $\boldsymbol{\sigma}_j^2$ is too large, then there will increased bias. In this case some basis functions effectively do not discriminate between input patterns. An intermediate choice of radial basis function $\boldsymbol{\sigma}_j^2$ balances minimises bias. A hierarchical k-fold cross validation determines the optimal number of basis functions in a first pass and the optimal centre positions and variances in a second pass.

A hierarchical 5-fold cross validation of a radial basis function correction of residual calibration errors of an affine stereo model involves the following steps:

[17] Examination of the horizontally straight plastic shielding for electrical cabling behind the RTX in Figure 5.6 reveals that its image is curved upwards away from the $Y-$ axis.

1. 5-fold cross-validated identification of the optimal radial basis function parameter order, i.e. the optimal number H of radial basis functions and optimal $\boldsymbol{\sigma}_j^2$.

2. 5-fold cross-validation of the optimal fit of optimal radial basis functions identified in the previous pass on an extended data set.

The final regression model prediction is an average of the outputs of an ensemble of estimators determined in the second cross validation pass. From the previous section, the data set $\mathbb{D}$ is divided into $m = 5$ pieces $(V_1, \ldots, V_m)$. Each V_i contains 25 elements. $V_{-i} = \mathbb{D} \backslash V_i$. Each V_{-i} contains 100 elements. In order to cross validate the radial basis function models, each V_{-i} is in turn split into five equal partitions V_j^{-i} of twenty elements, i.e. five sets of $V_1^{-i}, V_2^{-i}, V_3^{-i}, V_4^{-i}, V_5^{-i}$. For $j = 1 \ldots 5$, $V_{-j}^{-i} = V_{-i} \backslash V_j^{-i}$ is used to calibrate a network and V_j^{-i} is used as a validation set.

In order to estimate optimal $(\boldsymbol{\sigma}_j^2, H)$ in the first pass, it is first useful to determine a search range for H, the number of basis functions. A nonparametric regression heuristic is that the number of samples divided by the number of parameters per basis function leads to adequate nonparametric regression. To predict x, y, z coordinates corrections, a model with H radial basis functions has $6H$ parameters. For the first cross validation pass on 100 calibration data points with three outputs, i.e approximately 300 samples, the application of this heuristic generates a estimate of 40 basis functions as maximal upper-bound on the cross validated regression model search. Thus a cross-validation experiment is defined to search for optimal H in a range from 5 to 40 in steps of 5. In the general case, $\boldsymbol{\sigma}_j^2$ may be individual kernel j parameterised elliptical functions. However to simplify the problem, a single $\boldsymbol{\sigma}^2$ spherical variance is used for all basis functions with a radius parameter. To determine a reasonable range for the radius parameter of $\boldsymbol{\sigma}^2$, a useful heuristic is to estimate the average inter-sample distance. This can be evaluated by computing the squared distance of each data point to its nearest neighbour followed by averaging of these values to determine $\bar{d}^2$. This is done separately for each data set V_{-i}. $\bar{d}^2$ computed in this manner varies from 0.0013 to 0.0018 over different V_{-i}. The operation of the cross validation algorithm revealed that $\boldsymbol{\sigma}^2$ should be in a range close to 0.06, which is the approximate average random centre distance to its nearest neighbour centre. So $\bar{d}^2$ was set to 0.06. The search range of $\boldsymbol{\sigma}^2$ was set to $\bar{d}^2$ values of $0.2\bar{d}^2, 0.5\bar{d}^2, \bar{d}^2, 2\bar{d}^2, 5\bar{d}^2$. In order to determine the optimal order nonparametric model, it is necessary to do a five loop deep numerical experiment:

for $i = 1, \ldots 5$ (for V_i)
  for $a = 5, 10, \ldots, 50$ (basis functions H)
    for $b = \bar{d}^2$ values of $0.2\bar{d}^2, 0.5\bar{d}^2, \bar{d}^2, 2\bar{d}^2, 5\bar{d}^2$
      for $j = 1, \ldots, k = 5$
        for $l = 1, \ldots r = 5$ (for randomisation of basis function centres)
          Randomly select basis function centres.
          Calibrate the radial basis functions on V_{-j}^{-i}.
          Evaluate the validation error on V_j^{-i} to obtain $E_{i,j}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$.
        end (for l)
      end (for j)
      Determine $\bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ over the $k \times r$ experiments.
    end (for b)
  end (for a)
  Determine $\bar{E}^{c,d}(\boldsymbol{\sigma}_d^2, H_c) \leq \bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ for all a, b (see Table 5.9 and Table 5.11).
  for e = $1, \ldots 5$
    Calibrate model e with structure $\boldsymbol{\sigma}_d^2, H_c$ and different random centre placement on V_{-i}.

end (for e)

Determine the average error of the e models on V_{-i}.

Determine the average error of the e models on V_i.

end (for i)

i	optimal H	optimal σ^2	RMSE(V_{-i})	RMSE(V_i)
1	20	0.012	0.0658	0.0412
2	20	0.012	0.0686	0.0283
3	20	0.06	0.0583	0.0284
4	20	0.06	0.0427	0.0552
5	20	0.06	0.0488	0.0492

Table 5.9: Cross Validated Hyper-parameter Optima of the Radial Basis Function Correctors on V_i.

Table 5.9 presents the absolute error statistics of the ensemble e of estimators with the corresponding optimal variance and number of basis functions calibrated on V_i. The optimal validation set error is minimal for V_2. Thus the optimal nonparametric regressor to correct the affine stereo model has 120 parameters compared to the 12 parameters of the affine stereo model. While the procedure is completely automatic, it is computationally expensive. Of the order of $6250 = 5 \times 10 \times 5 \times 5 \times 5$ singular value decompositions were required to calibrate the radial basis function linear parameters during the 6250 calibration cycles. The absolute error statistics of the average of e estimators with $\sigma^2 = 0.012$ and $H = 20$ and different random centre placement calibrated on V_{-2} are presented in Table 5.10. The mean absolute error on the calibration set is $8.05cm$. This is $31.9cm$ better than the

	Calibration Error on V_{-2}			Test Error on V_2		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0678	0.1431	0.1732	0.0305	0.0749	0.0892
mean	0.0103	0.0323	0.0379	0.0100	0.0302	0.0354
variance	0.000000014	0.000000474	0.000001778	0.000000001	0.00000021	0.00000025

Table 5.10: Absolute Error Statistics of the Radial Basis Function Correction on V_2

mean absolute error of the optimal stereo-camera affine transform. On the validation set, the mean absolute error is $7.56cm$. This is $17.12cm$ better than the mean absolute error of the optimal stereo-camera affine transform. See Table 5.6. Figure 5.15 is a plot of the error of the optimal radial basis function ensemble corrector on the data set. Figure 5.16 is a histogram of the errors of the optimal radial basis function ensemble corrector. Figure 5.17 is a plot of the error of the optimal radial basis function ensemble corrector and the error of the optimal stereo-camera affine transform. Figure 5.18 is a scatter plot of the error of 125 data points of $\mathbf{S}_3$ against the optimal radial basis function corrector. Examination of Table 5.10 reveals that maximum $y = 14cm$ and $z = 17cm$ error on the calibration set are very high compared to the validation set. The mean accuracy of the optimal radial basis function correction is low compared to the results reported by (Walter and Schulten, 1993) (see §5.3.2) and in particular are far in excess of the camera quantisation resolution of approximately $1cm$ (see §6.1.2.1). The low mean absolute accuracy of the nonparametric correction and this discrepancy may indicate

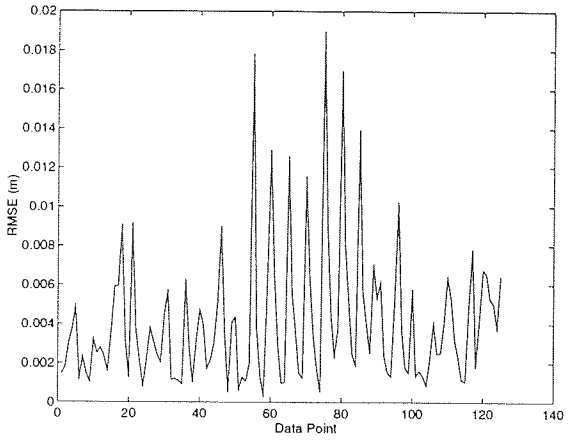

Figure 5.15: A RMSE Plot of the Optimal Radial Basis Function Correction of S_3.

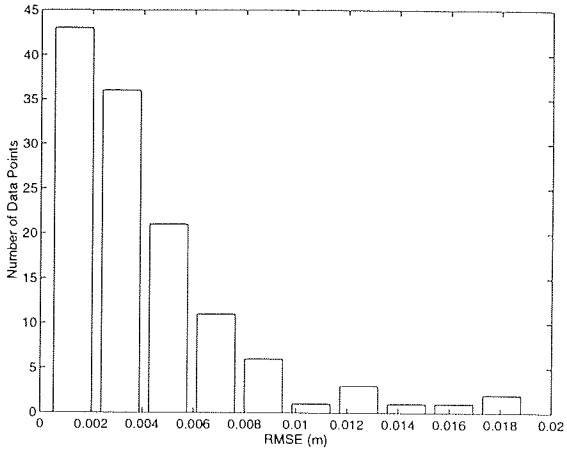

Figure 5.16: A Histogram of the RMSE of the Optimal Radial Basis Function Corrector.

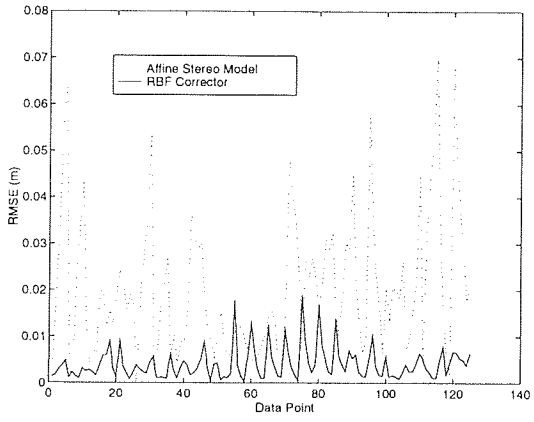

Figure 5.17: A RMSE Plot of S_3 and of the Optimal Radial Basis Function Correction of S_3.

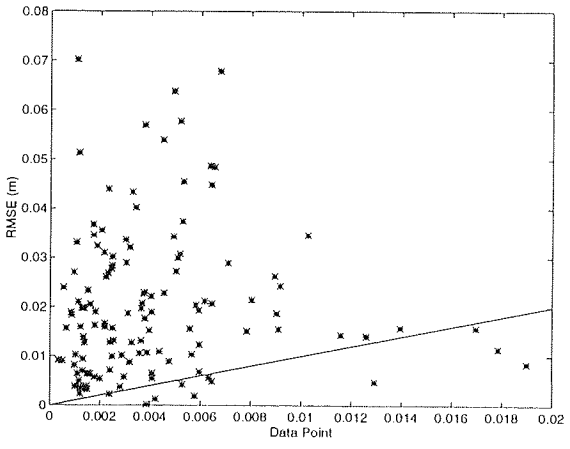

Figure 5.18: A Scatter Plot of the RMSE of S_3 against the RMSE of the Optimal Radial Basis Function Corrector.

the presence of an outlier in the data set. An outlier in the data set could perhaps have occurred as a result of a failure of the LED segmentation algorithm on one or more images. An outlier can seriously distort a cross validated regression. In order to deal with outliers it is necessary to do robust regression. Robust regression methods evaluate the possibility of outliers by leaving out data subsets and monitoring the effect on regression. A statistical difference in regression error for a given data subset indicates the presence of an outlier. See (Seber and Wild, 1989). Since a robust cross validated regression would increase the computational complexity of the regression experiment by two to three order of magnitude, (i.e. requiring 625,000 to 6,250,000 singular value decompositions) it was not undertaken. This simple cross validation regression experiment could thus be extended to make it more robust. The regression experiment could also be extended by utilisation of more complex basis function parameter estimation algorithms such as Expectation (conditional likelihood) Maximisation (Dempster et al., 1977).

Alternatively, the low accuracy may be due to the small number of samples used to approximate a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ mapping, 125 in contrast to 10,000 or (Walter and Schulten, 1993). This could be checked by increasing the sample density. This is not done here since the procedure is a demonstration of hierarchical cross validation not a proposed application since sequential parametric regression is simple. See Chapter 3.

To demonstrate that a sequential linear and nonparametric nonlinear regression has advantages over a pure nonlinear regression, a radial basis function approximation of the complete mapping from stereo-image feature measurement space to joint space was also undertaken by an obvious modification of the numerical experimental loop on pg. 98. In the case of a complete mapping the number of parameters per basis function is $7H$, hence a basis function range of $5, 10, \ldots, 50$ was searched. Table 5.11 presents the optimal number of basis functions and spherical variance parameters for a complete hierarchically cross validated radial basis function ensemble approximation of the mapping.

V_3 has the lowest cross validation error. The maximum, mean and variance of the absolute error residuals of hierarchically cross validated radial basis function ensemble approximations of the complete mapping are tabulated in Table 5.12, Table 5.13, Table 5.14, Table 5.15 and Table 5.16.

For the optimal ensemble estimators calibrated on V_3, the mean absolute error on the calibration set is $13.34cm$. This is $26.21cm$ better than $\mathbf{S}_2$ and $5.29cm$ worse than the radial basis function ensemble corrector. On the validation set, the mean absolute error is $16.76cm$. This is $7.92cm$ better than $\mathbf{S}_2$ and $9.2cm$ worse than the radial basis function ensemble corrector. Figure 5.27 is a scatter plot of the error of the optimal nonparametric ensemble approximation of the complete mapping against the error of the optimal nonparametric ensemble affine stereo-model correction. Figure 5.28 is histogram of the differences between the above errors. Both these plots clearly show that the correction approximation has on average higher accuracy than the complete approximation. Thus the cross validated radial basis function ensemble approximation of the complete mapping is better than the optimal affine transform stereo-model but worse than the cross validated radial basis ensemble correction of the optimal affine transform stereo-model. The latter regression model also has less parameters (124) than the complete mapping approximation ensemble ($175 = 25 \times 7$).

Figure 5.19 is a plot of the error of the optimal cross validated ensemble approximation of the complete mapping. Figure 5.20 is a histogram of the errors of the complete optimal nonparametric approximation. Figure 5.21 is a plot of the error of the complete optimal (determined on V_3) nonparametric approximation and the optimal affine stereo-model approximation. Figure 5.22, Figure 5.23, Figure 5.24, Figure 5.25 and Figure 5.26 are scatter plots of the error of the optimal affine stereo-model against each of the five optimised ensemble nonparametric estimators.

i	optimal H	optimal σ^2	RMSE(V_{-i})	RMSE(V_i)
1	25	0.6	0.0952	0.1447
2	25	0.6	0.0986	0.1284
3	25	0.6	0.0997	0.1203
4	25	0.6	0.1002	0.1262
5	24	0.6	0.0992	0.1443

Table 5.11: Cross Validated Hyper-parameter Optima of the Radial Basis Function Complete Approximation of V_i.

	Calibration Error on V_{-1}			Test Error on V_1		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0708	0.1760	0.1003	0.0563	0.2091	0.1003
mean	0.0233	0.0616	0.0439	0.0343	0.0970	0.0601
variance	0.0003	0.0016	0.0010	0.0004	0.0052	0.0015

Table 5.12: Absolute Error Statistics of the Radial Basis Function Complete Approximation of V_1.

	Calibration Error on V_{-2}			Test Error on V_2		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0714	0.2066	0.1017	0.0511	0.1735	0.1017
mean	0.0240	0.0626	0.0451	0.0298	0.0871	0.0603
variance	0.0003	0.0020	0.0009	0.0003	0.0029	0.0014

Table 5.13: Absolute Error Statistics of the Radial Basis Function Complete Approximation of V_2.

	Calibration Error on V_{-3}			Test Error on V_3		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0752	0.1918	0.1021	0.0493	0.1336	0.1021
mean	0.0234	0.0646	0.0454	0.0276	0.0796	0.0604
variance	0.0003	0.0019	0.0010	0.0002	0.0022	0.0014

Table 5.14: Absolute Error Statistics of the Radial Basis Function Complete Approximation of V_3.

	Calibration Error on V_{-4}			Test Error on V_4		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0234	0.0646	0.0454	0.0276	0.0796	0.0604
mean	0.0231	0.0645	0.0451	0.0316	0.0841	0.0605
variance	0.0003	0.0021	0.0009	0.0005	0.0025	0.0014

Table 5.15: Absolute Error Statistics of the Radial Basis Function Complete Approximation of V_4.

	Calibration Error on V_{-5}			Test Error on V_5		
Error	$x(m)$	$y(m)$	$z(m)$	$x(m)$	$y(m)$	$z(m)$
max	0.0762	0.1827	0.1004	0.0993	0.1926	0.1004
mean	0.0210	0.0638	0.0463	0.0424	0.0937	0.0601
variance	0.0003	0.0019	0.0011	0.0012	0.0042	0.0014

Table 5.16: Absolute Error Statistics of the Radial Basis Function Complete Approximation of V_5.

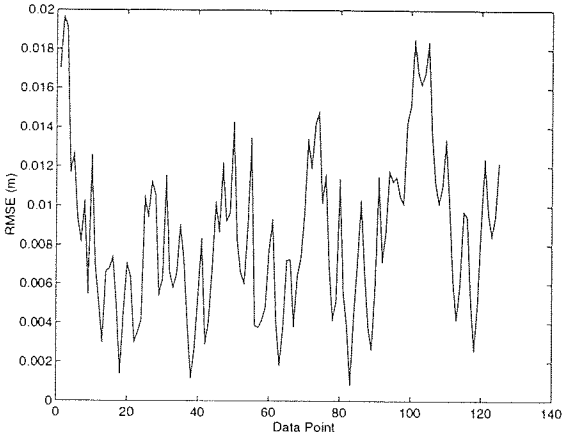

Figure 5.19: The RMSE of the Optimal Radial Basis Function Approximation of the Complete Mapping.

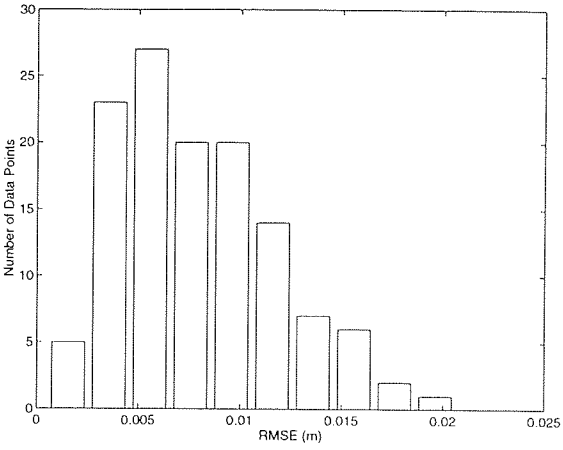

Figure 5.20: A Histogram of the RMSE of the Optimal Radial Basis Function Approximation of the Complete Mapping.

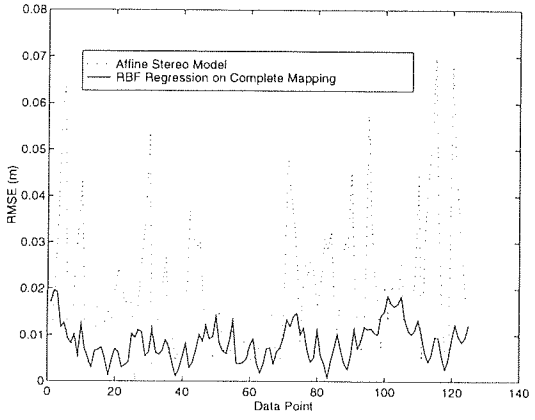

Figure 5.21: A RMSE Plot of S_3 and of the Optimal Radial Basis Function Approximation of the Complete Mapping.

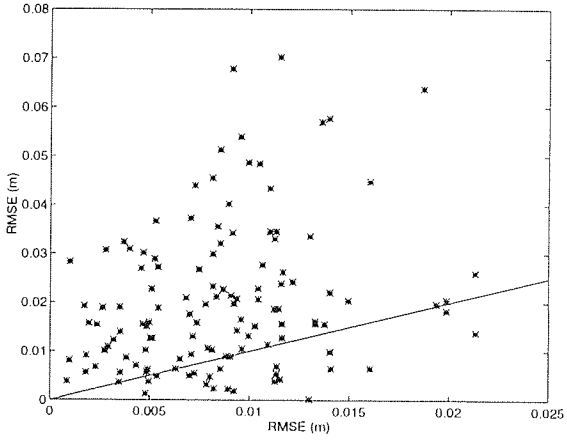

Figure 5.22: A Scatter Plot of the RMSE of S_3 against the RMSE of the Radial Basis Function Approximation, Determined on V_1, for the Complete Mapping.

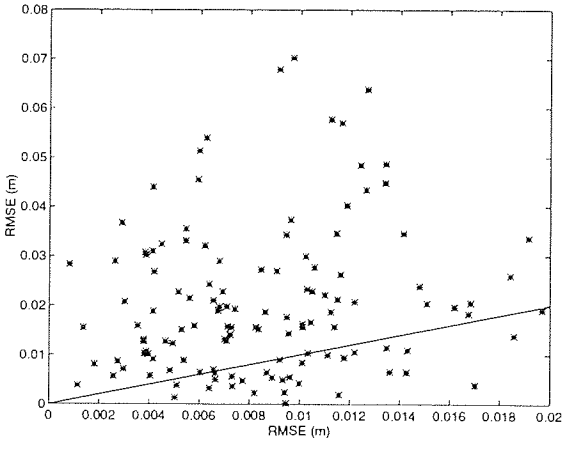

Figure 5.23: A Scatter Plot of the RMSE of S_3 against the RMSE of the Radial Basis Function Approximation, Determined on V_2, for the Complete Mapping.

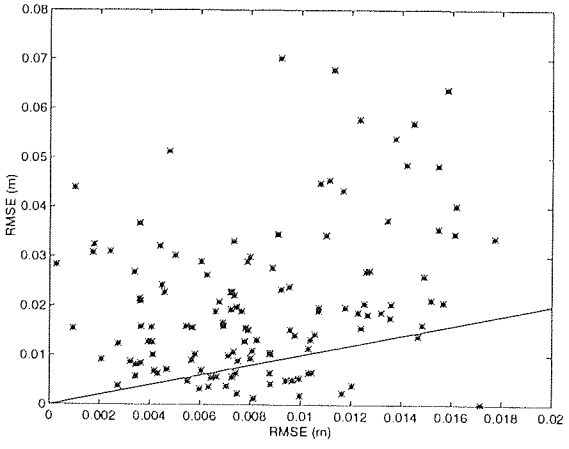

Figure 5.24: A Scatter Plot of the RMSE of S_3 against the RMSE of the Radial Basis Function Approximation, Determined on V_3, for the Complete Mapping.

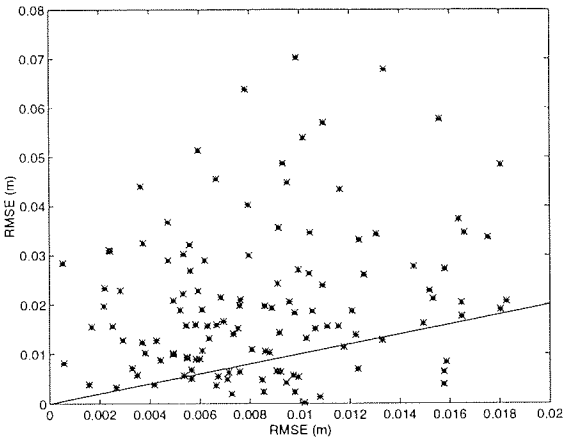

Figure 5.25: A Scatter Plot of the RMSE of S_3 against the RMSE of the Radial Basis Function Approximation, Determined on V_4, for the Complete Mapping.

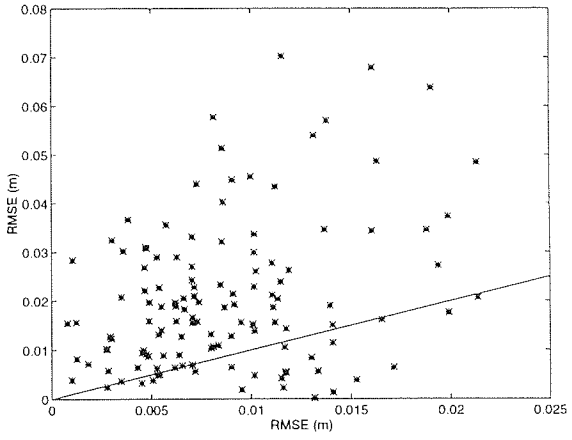

Figure 5.26: A Scatter Plot of the RMSE of S_3 against the RMSE of the Radial Basis Function Approximation, Determined on V_5, for the Complete Mapping.

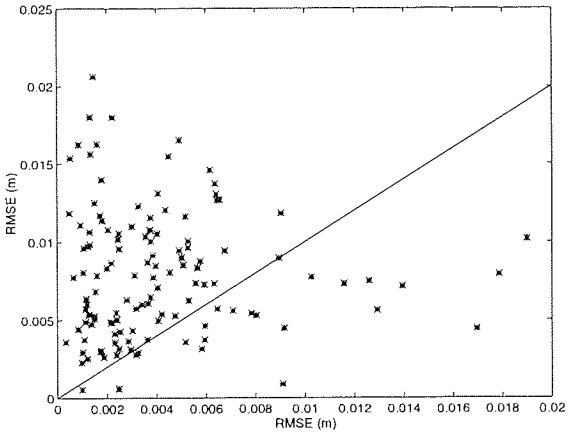

Figure 5.27: A Scatter Plot of the RMSE of the Optimal Radial Basis Function Approximation of the Complete Mapping against the RMSE of the optimal Radial Basis Function Corrector.

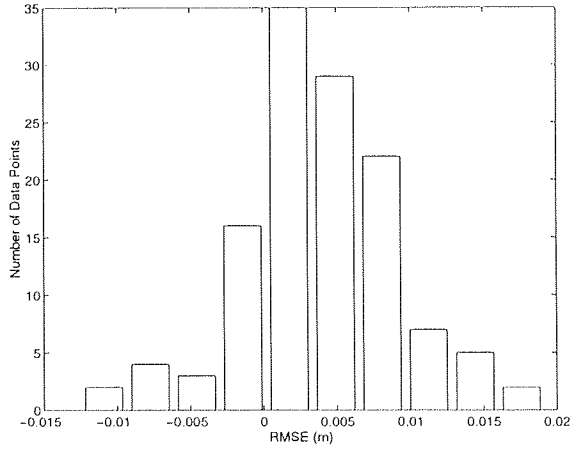

Figure 5.28: A Histogram of the RMSE of the Optimal Radial Basis Function Approximation of the Complete Mapping minus the RMSE of the Optimal Radial Basis Function Corrector.

The above procedure is a simple general completely automatic black box procedure that could be applied by technicians to low dimensional regression problems at higher levels of robotics and automation systems if analytical analysis fails. This is not the case for calibration of cameras to manipulators. See Chapter 3. However the principle function of the implementation of hierarchical cross validation of regression was to validate the distributed object manipulator control environment developed in the next Chapter.

Chapter 6

Distributed Object Manipulator Control Systems

Physical manipulators and cameras are controlled by complex software. In order to prototype distributed object software concepts and architectures for hierarchical control of distributed cameras and robot manipulators, a 5,000 line C++ distributed object environment was specified, implemented, and validated with the assistance of (Amador, 1995; Ng, 1995). Object oriented data abstraction and encapsulation were employed to implement the environment in order to cleanly separate programming of a software validation calibration experiment from programming of manipulator and sensor device control details. As a result of implementation experience, a methodology for system analysis, specification and design is proposed. In addition, fundamental non-functional requirements for scalable virtual micro-kernel brokers of virtual communication events in real-time functional objection composition hierarchical distributed systems are defined.

6.1 A Distributed Object Environment for Prototyping of Manipulator Control Algorithms

Most aspects of nonparametric regression, camera, robot kinematic and dynamic models except for real-time can be conveniently[1] represented and manipulated in Matlab. Matlab problem and algorithm representation cleanly separates mathematical robot algorithms from peripheral machine dependent I/O for signal acquisition and graphical display. Matlab representations and algorithms are mapped to communication protocols, operating systems, interfaces, data structures, and compilers by a socket-based multi-process object oriented manipulator and camera control environment.

The basic object system development construct of modular transformations of encapsulated data sets is extended by object oriented language syntactic constructs that reduce the length of application code expression by calls to object libraries. Object languages enable developers to:

1. define inheritance of classes,
2. define polymorphism of function naming over different data types,
3. define composition of classes,

[1] The (Corke, 1994) robotics toolbox was used for simulation of manipulators.

4. define operator overloading over different data types.

The distributed object manipulator control environment's three levels of software abstraction correspond to (Marr, 1982)'s three loosely coupled functional levels for modelling vision:

6.1 Definition (The Computational Specification Level)
This level characterises the function of a computation. This level is equivalent to a mathematical description of processing in terms of for instance recursively enumerable functions.

6.2 Definition (The Representation and Algorithm Specification Level)
A computational function is realised by an input output representation and transformations at an algorithm level. The programming language chosen to express the mathematical computation has certain semantic and syntactical constraints.

6.3 Definition (The Implementation Level)
A physical implementation hardware level realises an algorithm that realises a computable function.

The environment's three levels of software abstraction are:

❶ A Matlab algorithm is close to a computational level description that is separated from device dependencies.

❷ Device encapsulation C++ classes implemented by (Amador, 1995) wrap specific device control software.

(a) generic open-chain manipulator kinematics - relatively straightforward classes.

(b) a subset of generic image processing and image display - complex classes.

(c) a generic virtual client-server interface for communicating sequential processes. A derived class was implemented for UNIX sockets with message encoding/decoding into machine independent representation - intermediate complexity classes.

❸ Specific low level C code controls hardware.

(a) XIL C library calls are used to control a Sun-Video board.

(b) 3 functional Layers of C code for low level kinematic control of the RT100.

(Ng, 1995) designed and implemented a graphical interface for remote control of a RT100 manipulator. See Figure 6.1. (Ng, 1995) extended this to stereo-image point and click based kinematic tele-command. See §6.1.4.

6.1.1 The Manipulator and Imaging Hardware

The following hardware was integrated into the development environment:

- 1 RT100 SCARA manipulator,
- 2 Sun-Cameras,
- 1 Sun-Video board,
- 1 red LED feature held by a 1 *dof* gripper,
- 1 networked SPARC classic **zippy**,

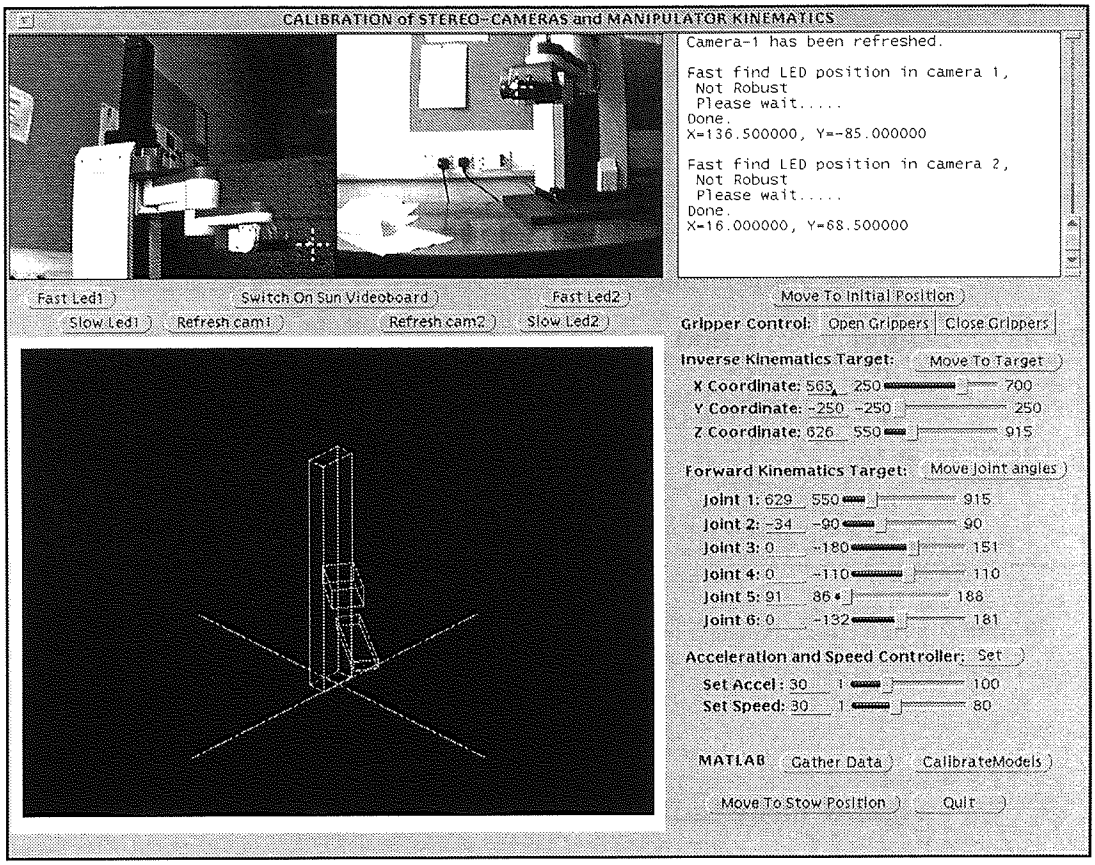

Figure 6.1: The Graphical Tele-command Interface.

- 1 networked SPARC10 **synapse** compute server on which Matlab executes.

A SCARA robot has a simple kinematic structure. SCARA design simplifies kinematic and dynamic analysis of interaction of payloads, friction and vibration but there are operational holes in a workspace $\mathfrak{W}$. Table 6.1.1 presents the first three *dof* derived from examination of (Universal Machine Intelligence Ltd., 1987). $d_3 = -15.0cm$ is based on measurement with a ruler since this constant is not indicated in (Universal Machine Intelligence Ltd., 1987). $\theta_4 = 0^o, \theta_5 = \pi/2, \theta_6 = 0$ corresponds to a fixed gripper pointing straight out. The lower arm measurement incorporates distances:

- from wrist to the tool centre point of the grip is 17.3cm,
- from grip to tip is 0.9 cm,
- 1 cm from tip to centre of red LED.

A SCARA manipulator has simple kinematics. For the first three *dof*:

- $z = d_1 - d_3$,
- θ_1, θ_2 are inverse of planar manipulator of §2.5.1.

(Universal Machine Intelligence Ltd., 1987) specify that the nominal maximum controllable payload is $2kg$. The nominal ex-factory repeatability is $\pm 1.0cm$. The actual accuracy was empirically determined to be of the order $\pm 1.0cm$. The overall low repeatability, low precision motors, and low

payload capacity of the plastic gripper limit attainable precision of a calibration procedure. The joints of the RT100 are controlled[2] by two internal Intelligent Peripheral boards - IP0 and IP1. IP0 and IP1 are controlled by an Intelligent Peripherals Communication Protocol. Kinematic commands are issued by a SPARC classic **zippy** via an RS232 cable. The basic communication atom between the control computer and the manipulator device is a frame. A frame is composed of a 1-2 byte command from the control computer followed by a 1-3 byte response from the device containing status information related to command processing. Command and status responses are exchanged over RS232 in an 8ms frame. The communication protocol is sequential frame command of IP0 and IP1.

The two auto-focus Sun-Cameras were mounted on pan-tilt tripods. The Sun-Camera lenses have a relatively large wide angle distortion. See Figure 5.6. The CCD photon collector of each camera measures $5.14cm \times 4.9cm$ and has a $0.1mm$ pixel resolution. The two Sun-Cameras output NTSC format images to a Sun-Video board. The Sun-Video board can be programmed with X Image Library (XIL) C library functions to perform image processing operations.

link	joint	twist	length	offset
i	θ_i	α_i	a_i(cm)	d_i (cm)
1	0	0	0	d_1
2	θ_2	0	25.35	0.0
3	θ_3	0	0	-15.0
4	0	0	25.35+17.3+0.9+1.0	0

Table 6.1: The Denavit Hartenberg Table of a 3 *dof* RT100.

6.1.2 An Algorithm for Segmentation of a Red LED Image

Two Sun-Cameras measure the position of the RT100 manipulator in image space (See Figure 5.5, Figure 5.6). Table 3.1 indicates the number of non-collinear point features required by sets of cameras to uniquely specify position and orientation of a manipulator in SE(3). With stereo-cameras, a single feature is sufficient for calibration of DH parameters governing position of a manipulator. To simplify but not trivialise[3] image processing, power wiring and a switch were attached[4] to IP1 in order to switch on/off a red LED held by the RT100 gripper. (Amador, 1995) designed and implemented C++ classes for:

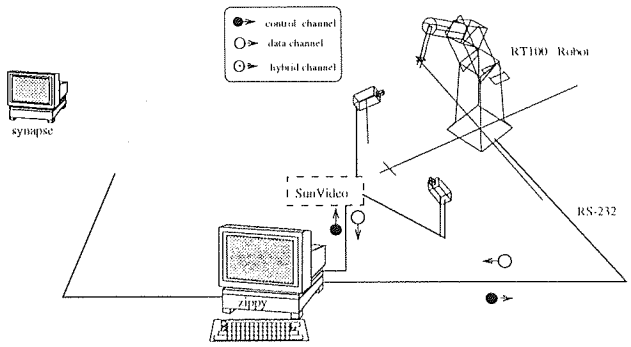

Figure 6.2: The Hardware Topography.

1. capture of stereo-images,

2. LED segmentation:

   (a) computation of stereo-image contents $\{a, b, c \ldots\}$ = objects_in_image().

[2]The manufacturer's original PASCAL/DOS code for controlling these two boards was replaced by C code that was ported to execute on a UNIX control computer.

[3]For instance by using an infra red LED, a homogeneous background, and artificial DC illumination. See (Horn, 1986; Vernon, 1991) for a discussion of design factors in construction of industrial imaging environments to simplify inspection image processing algorithmic complexity. The 50hz AC fluorescent strip light in the room creates high contrast switching noise that is picked up by the CCD but not the human eye. The window was deliberately not curtained to challenge the LED segmentation algorithm. A passing cloud reduces image intensity by up to 120 grey levels in a 256 grey level intensity scale. The segmentation algorithm is robust to all of these effects.

[4]I gratefully acknowledge the assistance of Andrew Sutton for help with re-wiring the IP1 board.

(b) computation of stereo-image content locations.
$\{a_x, a_y, b_x, b_y, c_x, c_y \ldots\}$ = location_in_image($\{a, b, c \ldots\}$).

3. image processing operations:

   (a) arithmetic_process(image_type image) to compute image differences and thresholding.

   (b) geometric_process(image_type image) to draw lines on images, and cluster pixels.

   (c) colour_process(image_type image) to operate on different combinations of `Red-Green-Blue` (RGB) and `Hue-Saturation-Intensity` (HSI) image representations.

   (d) convolution_process(image_type image) filters such as Gaussian and low pass filter.

4. general tele-image manipulation:

   (a) export_image(image_type image, format new_format) for conversion between HTSC and X11 colormaps.

   (b) encode_image(image_type image) into machine independent representations.

   (c) transmit_image(image_type image) over sockets.

   (d) compress_image(image_type image) uusing a SunVideo board data compression operation.

   (e) de-compress_image(image_type image) reverse operation.

   (f) decode_image(image_type image) into specific machine representation.

   (g) display_image(image_type image) on X11 and eliminate colormap switching.

While the calibration experiment environment is partially controlled by removal of most reddish objects apart from the LED, information contained in image histograms must be combined with a priori modelling for LED segmentation. A red LED segmentation algorithm was developed that is robust to noisy natural illumination of a large window next to the RT100. Algorithms for feature segmentation in an image are generally based on a combination of (Fu et al., 1987; Horn, 1986):

- Measurements of optical flow if sensors are active or features can move,
- Edge detection, for instance by application of a Laplacian of Gaussian filter,
- Feature region morphing growing, splitting, and merging,
- Thresholding of intensity histograms.

A maximum threshold of the red band of a RGB image is insufficient to reliable segment a red LED. (Amador, 1995) developed an algorithm for LED segmentation that combined thresholding and feature cluster merging. If $\mathcal{I}$ is an image, $\{p_{x,y} | \forall p_{x,y} \in \mathcal{I}, t^l \leq p_{x,y} \leq t^u\}$ is a set of pixels, where t^l is a lower bound and t^u is an upper bound on application of the inequality operations to an image band, for instance the red band. t^l, t^y can be determined in a heuristic manner as $t^u = \max(p_{x,y}) - a$, $t^l = \min(p_{x,y}) + b$ or $t^l = t^u - \frac{t^u - t^l}{b}$ where possibly $a = f_1(\mathcal{I}), b = f_2(\mathcal{I})$. It was determined that setting $a = 0$ and $t^l = t^u - 1$, i.e. finding the reddest pixel did not always segment the LED because of natural illumination variation and matt red objects in the environment.

A measure of LED redness based on both RBG and HSI colour spaces, was empirically determined by (Amador, 1995) to be

$$\text{Red}_{\text{LED}} = \frac{\text{Red} - (\text{Green} + \text{Blue})}{2} - \text{Hue} \tag{6.1}$$

where Hue (see (Horn, 1986)) is an angular colour representation with Red = $0/2\pi$, with 2π used, is thresholded, segmentation of the LED is more reliable but still not 100% reliable. In order to increase segmentation reliability, a priori knowledge can be incorporated into a feature region morphing algorithm. (Amador, 1995) empirically determined that the number of red LED pixels varied between 10 and 500 depending on distance of the LED from cameras for calibration experiments. Histogram analysis determined a t^l to extract 500 reddest pixels.

(Amador, 1995)'s analysis of sets of Red_{LED} scale images indicate that LED images are composed of high intensity regions separated by lower intensity holes. Also, no sharp intensity gradient at LED edges exists that could be detected with for instance a Hough transform for extraction of parameterised red LED boundary lines initially hypothesised to be present in images. After review of the literature on blob segmentation, (Amador, 1995) implemented a modification of the (Chen, 1994) algorithm for pixel cluster detection to segment the LED. The modified (Chen, 1994) algorithm has the following operations:

1. The hybrid image space Red_{LED} is thresholded with the empirically determined values for extraction of the LED.
2. Solitary pixel noise is removed from the thresholded image.
3. This leaves connected thresholded pixel sub-clusters.
4. An adjacency matrix of the sub-clusters is constructed.
5. The mean distance of a pixel sub-cluster to its nearest neighbour pixel sub-cluster is computed.
6. Pairwise sub-clusters with small mean inter-cluster distance are merged into super-clusters and the adjacency matrix is updated.
7. This computation is iterated until a predetermined maximum inter-cluster distance is reached.
8. The final set of non-oriented graphs extracted from the adjacency matrix is a set of detected clusters.
9. The LED cluster is extracted by computing the mean red value for each cluster and selection of cluster with highest mean Red_{LED} value.
10. The image coordinates of the LED are computed as the centroid of the LED cluster or the reddest pixel. These values are empirically similar.

This algorithm reliably segments the RED led from the images. The stereo-images are displayed in 2 X-windows of the GUI. Crosses are added to the images to indicate the centre of the estimate of LED position in the image. See Figure 6.1.

The modified (Chen, 1994) pixel cluster detection algorithm reliably segments the LED from a naturally illuminated background which contains matt red objects. When segmentation fails is a function of LED orientation relative to a camera image plane. The LED is a conical light emitter and when the manipulator positions the LED in certain side-on configuration relative to a camera image plane, the emitted cone of light is far from the camera image plane and the LED segmentation algorithm fails.

LED size may be estimated by the number of cluster pixels, or standard x, y deviation of cluster pixels. LED size estimates are sensitive to the orientation of the LED light cone. Depending on the

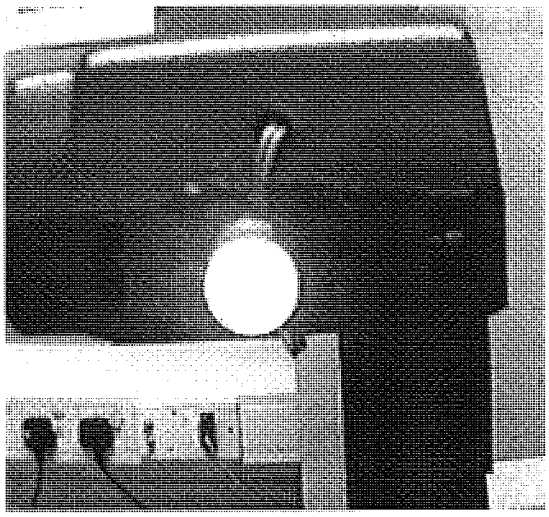

Figure 6.3: The LED at 10cm from the CCD.

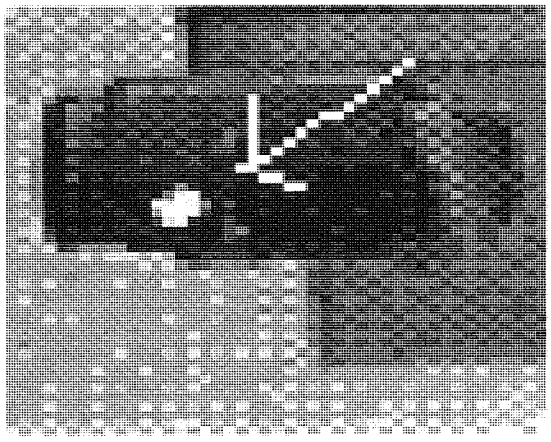

Figure 6.4: The LED at 1.5m from the CCD.

illumination of the room and the distance of LED from cameras, a variable LED halo[5] also introduces large variance into size measurement. See Figure 6.1.2. Hence size measurements are not used for single camera calibration based on a feature size model. Neither are marginal accuracy improvement techniques such as sub-pixel interpolation (Horn, 1986).

The difficulty of reliable LED segmentation from images containing fluorescent strip 50hz AC illumination strobing and natural illumination variation indicates the difficulty of reliably segmenting an image of the gripper itself without a relatively easy to segment feature such as a red LED.

6.1.2.1 The Precision of Image Measurements of the LED Position

The precision of measurements of LED position in $\mathbf{x} \in \mathfrak{C}$ that can be extracted from image measurements $\mathbf{x}'$ is a function of

- **systematic** interior f and exterior $\mathbf{A}_{1,2}$ model errors relating image measurements $\mathbf{x}'$ to $\mathbf{x}$.
- **noise** ϖ in image measurement. This is a function of
  - illumination variation, either natural or AC flicker.
  - analog to digital quantisation round off errors defined by

$$\mathbf{P} : \mathbb{R}^3 \rightarrow \mathbb{Z}^2 : \ \backprime\mathbf{x} \rightarrow \mathbf{x}' : x' = \lceil \frac{\varsigma f \backprime x}{\backprime z} \rceil, \tag{6.2}$$

$$\mathbf{P}^{-1} : \mathbb{Z}^2 \rightarrow \mathbb{R}^3 : \ x = \frac{x' \ z}{\varsigma f} \pm \varpi(z). \tag{6.3}$$

where $\lceil \ \rceil$ is rounding to the nearest integer determined by ς, a parameter for scaling measurements x', y' to an appropriate integer resolution of a CCD. ϖ is an input dependent noise function. Measurement accuracy is a function of distance $\backprime z$ of a manipulator feature from a camera. Steady cameras placed at $2m$ from the base of a manipulator typically have measurement precision of $5 - 10mm$ in $\backprime x, \backprime y, \backprime z$. In Figure 6.1.2, the LED is approximately $10cm$ from the lens. In Figure 6.1.2, the LED is approximately $1.5m$ from the lens. The pixel granularity shows that the image of Figure 6.1.2 is magnified compared to Figure 6.1.2 with consequent loss of pixel precision.

[5]Red light is close to infra-red heat emissions. The camera CCD is sensitive to the higher end of the infra-red spectrum. The halo around the camera measurements is due to heat detection. This is also the cause of the fuzzy LED image boundaries.

6.1.3 The Environment Control Flow

There are three[6] computational processes in the environment (see Figure 6.5):

① a manipulator control process executes on **zippy**.

② a graphic command interface and image processing process usually executes on **zippy**. GUI display executing on a remote Xserver, for instance **synapse**, implements a tele-manipulation interface.

③ a communication process that relays commands from `matlab` to ② executes on **synapse**.

② is a client of server ① through a socket s_1. ② is a client of server ③ through socket s_2. The three processes are connected in a linear fashion by two sockets ③$\xrightarrow{s_2}$②$\xrightarrow{s_1}$①. In order for ③ to control the RT100, ③ sends a command c_i over s_1 to ② which relays c_i over s_1 to ①. See Figure 6.5. The hardware control loop involves the following operations:

1. A shell command initiates ①, the robot controller process, on the **zippy** processor.
2. A shell command initiates ②, the GUI+imaging process (c.f. Figure 6.1), on the **zippy** processor.
3. A click of the Move To Initial Position button on the GUI creates the s_2 socket between ② and ③ and initialises the RT100.
4. A click the Switch On Sun Videoboard initialises image processing and display.
5. A click on the Matlab: Gather Data button initialises the s_2 socket from ② to ③, from the GUI process to a remote matlab controller process, and blocks further operation of the GUI.
6. matlab is initialised through an emacs buffer command interpreter with Emacs executing on the faster **synapse** processor.
7. From the matlab interpreted command line, the server software ③ is initialised and connected via s_2 to ②, the blocked GUI process. The hardware control loop is thus

$$\text{matlab} \longrightarrow \text{emacs} \longrightarrow ③ \xrightarrow{s_2} \{②+\text{stereocam}\} \xrightarrow{s_1} ① \rightarrow \text{RT100}$$

8. For a calibration experiment, e.g. in §5.5.1, a joint sample range is generated $\boldsymbol{\theta}_i, i = 1, \ldots, n$ and for each joint command $\boldsymbol{\theta}_i = \mathsf{c}_i$:
   (a) c_i is sent to ②.
   (b) ② receives c_i and forwards c_i to ③.
   (c) ③ receives c_i and instructs IP0 and IP1 to move the RT100 joints to c_i. See Appendix C.
   (d) When the RT100 has reached c_i and has finished moving, ① returns a c_i completion status message to ②.
   (e) ② activates the image capture and processing software. The pixel coordinates of the LED are extracted from the stereo-images and the target is identified in the images and displayed on the GUI.

[6] The environment supports process and socket communication replication. The classes were designed for multi-robot, multi-stereocamera control.

(f) ② sends[7] the stereo-image coordinate vector of the LED to ③ and thence to matlab

(Amador, 1995; Ng, 1995) implemented an error checking non-nominal decision tree to validate the interaction of the three processes in addition to each single process functioning. Non-nominal feedback between ①,②, and ③ is implemented by the C++ object oriented `throw` construct for hierarchical error checking and generation of inputs into an error recovery decision tree wrapped around the socket communication calls.

6.1.4 The Graphical Tele-command Interface to Manipulators and Cameras

(Ng, 1995) and I designed and implemented a graphical interface to the hardware control loop for remote control of the RT100 and SunCameras. The main ergonomic design requirement for the GUI was minimisation of the sum of average cursor travel distances between GUI commands weighted by expected frequency. The graphical interface displays the stereo-camera images in X11 colormaps, displays a robot Xlib wireframe, and supports robot and camera command. The Xlib-based robot wireframe can be zoomed, moved relative to the X,Y coordinate directions of the viewing plane, and rotated by ψ, ϕ pitch and azimuth relative to the viewing plane by a combination of keyboard and mouse commands. The robot wireframe C code was originally developed by (van der Smagt, 1994). (Ng, 1995) reverse and re-engineered this code for integration into the C++ robot control environment. See the wire-frame robot in the bottom left hand corner of Figure 6.1. The graphical interface supports commands for initialising and resetting robot and camera control board communications with the environment, command of the robot to move to Cartesian and joint space targets and processing of nominal and non-nominal diagnostic error output. See Figure 6.1. (Ng, 1995) extended this to stereo-image point and click based kinematic tele-command. See §6.1.4. (Ng, 1995) implemented wire-frame robot overlay on images and point and click indication of kinematic grasp targets and GUI slider bar for input of $\mathfrak{F}$ and joint target coordinates. See Figure 6.1.

If $\mathbf{PA}_1$ is a combined interior camera perspective projection and exterior affine translation where $\mathbf{x}' = \mathbf{PA}_1\mathbf{x}$, 6 samples are required to calibrate $\mathbf{PA}_1, \mathbf{A}_2$ from §3.1. Figure 6.6 shows $\mathbf{G}$ projected onto an image by uncalibrated $\mathbf{P}, \mathbf{A}_1, \mathbf{A}_2$. If $\mathbf{G}$ is a Cartesian vector wire-frame model of a manipulator, $\mathbf{PA}_1(\mathbf{G}), \mathbf{PA}_2(\mathbf{G})$ maps the Cartesian wire-frame model to its stereo image. $\mathbf{PA}_1(\mathbf{G}), \mathbf{PA}_2(\mathbf{G})$ project models of a manipulator onto stereo-images. This provides a check on calibration of $\mathbf{PA}_1, \mathbf{A}_2$ and it illustrates how solid geometry databases of objects in $\mathfrak{F}$ can be projected onto stereo images.

Projection of solid geometry models onto images forms a basis for many statistical pattern classification algorithms, for instance deformable model matching, for matching parametric object models to sensor data (Duda and Hart, 1973; Horn, 1986). Projection of solid models onto images also forms a basis for point and click image-based tele-command to move a manipulator or indicate object grasp targets when combined with suitable target selection peripherals and grasp planning and compliant force feedback software (see (Fu et al., 1987)). The simple[8] way to input SO(3) coordinates is via a joystick and to input SE(3) coordinates is via a rack of six twiddle knobs. For the purposes of environment prototyping, we decided to build a mouse interface to $\mathfrak{C}^3$ target selection to reduce the amount of command peripherals attached to the environment in order to satisfy the primary ergonomic GUI design criterion. (Ng, 1995) implemented mouse click on stereo-image based robot command. The

[7] $\mathfrak{c}_i$ round trip time from matlab to receiving the LED coordinates is approximately three minutes. Most of the $\mathfrak{c}_i$ execution time is consumed by image processing to find the LED. The two cameras are multiplexed into one Sun-Video board and software switching of input from camera 1 to camera 2 requires one minute.

[8] Cartesian command input peripherals can also be based on immersive graphical interfaces and glove sensors of hand and position finger vectors and velocities.

upper left corner of an image is the origin of X-window graphical coordinate system $\mathbf{O}_g$. The origin of the camera coordinate system is the middle of the image $\mathbf{O}_{im}$. Mouse coordinates input in $\mathbf{O}_g$ should be translated to $\mathbf{O}_{im}$. If one image is used, $\mathbf{P}^{-1}, \mathbf{A}_1^{-1}$ maps transformations of mouse coordinates from $\mathbf{x}'$ to $\{\mathbf{x}\}$. $\mathbf{P}^{-1}$ is an under-determined inverse. To select one coordinate in $\{\mathbf{x}\}$, a $\backslash z$ coordinate must be specified. $\mathbf{P}^{-1}$ generates a line of points projected onto an image. (Ng, 1995) implemented three methods to resolve a target on the single image projection to Cartesian space line:

1. z-slider bar based target resolution,
2. middle and right mouse button (in)de-crement of a $\backslash z$ scale,
3. a second mouse click on the projection to the second stereo-image of the Cartesian target line determined by the first click on image one.

Mouse point and left click input of $\mathbf{x}'_1, \mathbf{x}'_2$ target coordinates in stereo space requires accurate clicking on a corresponding feature after technician visual feature matching in the stereo-images. With unconstrained stereo-image clicking on the corresponding feature, $\mathbf{P}^{-1}, \mathbf{A}_{1,2}^{-1}$ project target position to $\mathfrak{F}$ coordinate system at $\mathbf{x}$, the intersection of projection axes or a probability function of the minimum distance between the two inverse perspective axes. The third target resolution method constrains the click on corresponding features by the technician to extract the depth of the grasp command. Once $\mathbf{x}$ is computed, $\mathbf{x}$ is represented in the calibrated wireframe by an XYZ coordinate axes icon. In Figure 6.1, this is the cross in the lower left corner of the image. The XYZ icon can also be projected to the stereo-images via $\mathbf{PA}_1, \mathbf{A}_2$ for display on each image for technician checking or iterative refinement of image based point and click target resolution. This basic GUI developed by (Ng, 1995) implements a kinematic substrate, for instance for Bayesian parametric pattern recognition based on a database of manipulation object centred representation hierarchy, for tele-manipulation of parametrically modeled objects. Parametric-model based pattern matching algorithms depend on good models of $\mathbf{P}, \mathbf{A}_1, \mathbf{A}_2$.

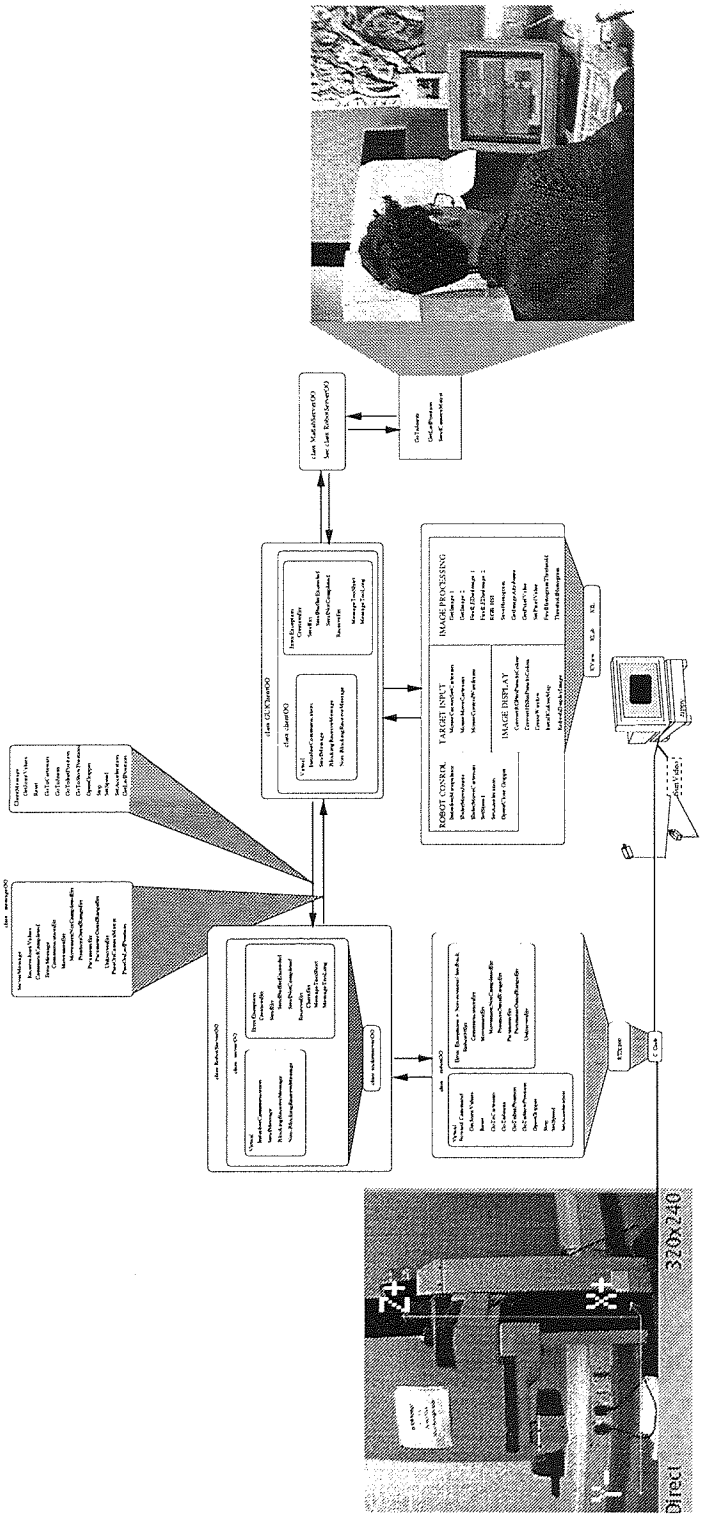

Figure 6.5: Organisation and Operation of the Distributed Object Environment.

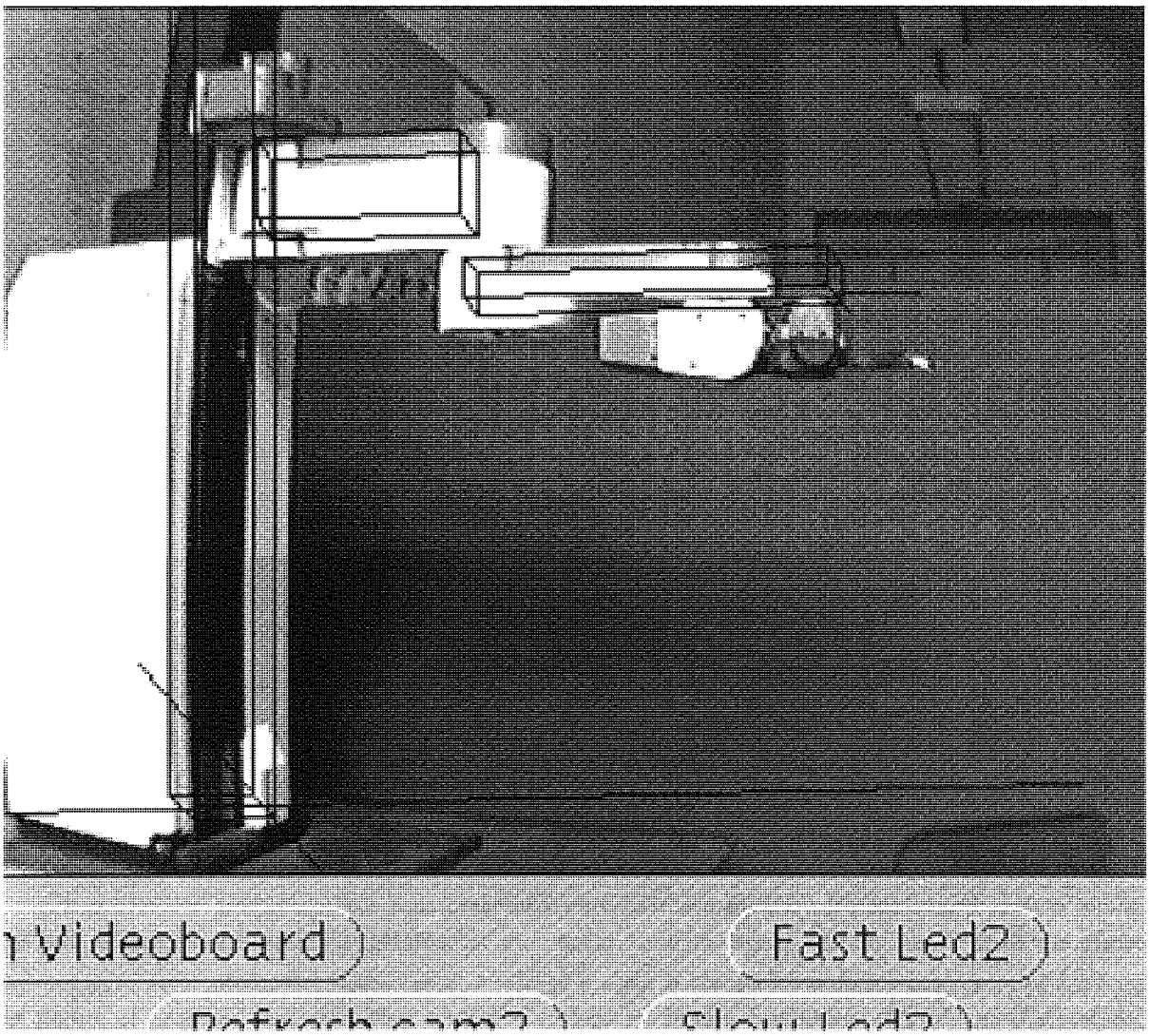

Figure 6.6: The Affine Transformation and Perspective Projection for Camera One.

6.2 Distributed Object Systems Integration Methodologies

Experience gained §6.1 at the C++ object implementation level proved useful to evaluate object system analysis and design methodologies. In order to design the distributed object environment for prototyping manipulator control algorithms, DFA (Elfving and Kirchhoff, 1991), NASREM (Albus et al., 1987), the Toaster and Booch methodologies were evaluated for match to the scope of prototyping requirements. Because of the simplicity of the prototype environment, the (Booch, 1994) object class inheritance and object interaction specification methodology was selected as the analysis and design notation for the environment. See (Amador, 1995; Ng, 1995).

Top down functional software engineering methodologies such as Design for Automation, see §1.1, are being gradually replaced by object system design and specification methodologies and architectures. The basic reason for adoption of object oriented software development is the limitation of short term memory and attention span of programmers. Bounded rationality requires organisation of large codes into modules. Bounded rationality also requires functional abstraction of object codes for analysis of dependability at the systems level. See §6.5.2.

Object oriented systems development evolved from and is similar to top down functional systems development. Both classes of analysis methods operate on a high level functional and non-functional requirements system specification. Functional methods sequentially define the data model and the data transformation model, i.e. the system functional structure. Object methods concurrently define a partitioned data model and data transformations that operate on the partitioned data. Object systems may have the following advantages over pure declarative, functional and logical software codes:

1. reusable components,
2. scaling of programming to large complex systems,
3. software maintainability,
4. more rapid application development.

if the system has been adequately analysed, designed, and implemented. Object system inheritance (with private, public, and protected data types) can reduce the length of code expression that calls object library functions, especially when combined with functional polymorphism, but inheritance (especially multiple) and function polymorphism make object systems difficult to analyse.

Object system characteristics, e.g. reusability, are more sensitive to the quality of system analysis and design than functional systems. It may occur that an object system has none of the above advantages, costs more and takes longer to develop than a comparative functional systems because of failure of problem analysis. This usually manifests itself as a restriction of problem analysis, i.e. a restriction of analysis of object operational contexts and context extensions. Object systems require a large amount of analysis with relatively little implementation of code. It is more difficult to learn object system analysis and design than functional systems analysis mainly because of the more complicated function call graphs generated by inheritance, polymorphism, and operator overloading syntactic constructs. Object system development may be more expensive in labour costs and time to system deployment in the short to medium term. Inheritance and polymorphism may delay overall development times if several developers are required to understand these structures in order to implement them. These constructs complicate debugging, analysis, verification, validation and re-engineering of software. If object languages are misused, object code developers can spend up to 80% of their time trying to understand their own and other developer's code in lieu of analysis and

design. If an object library is not subject to stringent verification and validation testing, the presence of a bug in an object system library can cause inordinate time wastage for application developers if they stumble into debugging of an object library polymorphic call graph.

Object system development projects with a consistently greater than 50% ratio of implementation time to analysis, specification, design, verification and validation phases are more likely to fail in the long run where failure is measured by non-reusability of components, escalating maintenance and bug fixing system development time ratios as code functionality increases. A secondary effect is that $\frac{f}{v}$ the ratio of code features/functions f to code volume v decreases until very large amounts of code volume must be generated to make minor feature extensions to the code base. In the worst case scenario, a medium to large scale badly analysed and developed object system may be impossible to reverse engineer or re-engineer compared to an equivalent size functional system because of the size of the polymorphic call graph if multiple inheritance has been liberally used. If this scenario occurs, it is cheaper in the long run to escrow the code base and reinitalise design and development since any extensions to a badly analysed system make the system even less analysable, increase instability of the code base and result in generation of more non-reusable fragile low performance code.

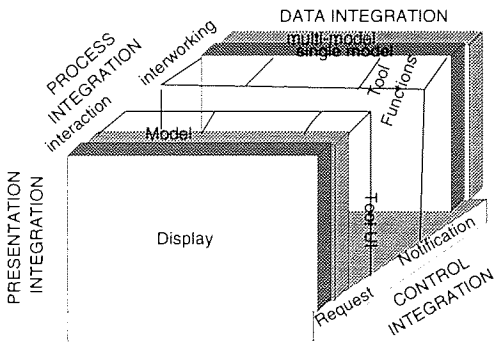

Figure 6.7: The ECMA/NIST Toaster Model of Software Integration Dimensions.

Quality object libraries accelerate software development and enable large amounts of functionality to be densely loaded into high level compact notations at the heart of system cores that can be rapidly and flexibly reconfigured to optimally process variable sets of system operational scenarios. Quality object libraries permit a developer to spend more time on requirements analysis, specification, and verification and validation with actual implementation work reduced to 25%-35% of project lifecycle.

Definition of a complete consistent distributed object integration system view is difficult because it is a greater than two dimensional specification problem. The European Computer Manufacturers Association/National Institute of Standards and Technology (ECMA/NIST) Toaster model is a valuable conceptualisation of dimensions of software integration. The Toaster model describes system services integration along four dimensions:

1. work process integration:
   (a) user-oriented work formalisation and methodology
      i. tools for data transformation,
   (b) group-oriented workflow, e.g. Petri net formalisation of workflow synchronisation.
      i. group tool use conventions.
2. data integration:
   (a) single model,
   (b) multi-model.
3. control integration:
   (a) request - synchronous operation,
   (b) notification - asynchronous operation.
4. presentation integration:
   (a) user-interface display look and feel,
   (b) common presentation of different functional models.

See Figure 6.2. The main integration dimensions of the NIST/ECMA Toaster model are three dimensional - data, workflow event and control integration. The four dimensional NIST/ECMA Toaster views of software logical organisation does not represent software distribution on processing meshes. It is with this fifth view that client-server application programmers are most familiar. If in addition other software organisation dimensions are incorporated for instance to represent software lifecycle, one quickly has a greater than six dimensional software engineering architectural space.

The NIST/ECMA software systems integration model does not neatly fit onto various architectural specification languages. An architectural language provides the system analyst with a language for specification of core system functionality and software organisation. Architecture languages (e.g. (Shaw et al., 1995)) are often graphical notations but they may also be formal syntactic specifications. An architectural specification describes:

1. system structure at an abstract computational level independent of implementation details,
2. key interactions of subsystem components,
3. global system properties such as non-functional dependability requirements.

(Booch, 1994) was useful to specify the environment but one of the limitations of (Booch, 1994) and other object analysis and design methodologies are that their notations tend to be close to the implementation level and they are thus unsuitable architectural notations. The experience of the prototype was that (Booch, 1994) notations tend to restrict developer search for optimal class hierarchies. The (Booch, 1994) notation (not the method itself) may lead to a developer focus on sub-classing design activities often related to particular object language advanced features instead of initial base class designs. An initial haphazard analysis of base classes may be quickly concretised by refinement of subclasses and implementation oriented features. A review of the software architectural research literature, e.g. (Shaw et al., 1995), did not reveal a satisfactory architecture design method or indeed a good notation. Hence a prototype architecture design methodology was defined.

6.3 A Methodology for Design of a Functional Object Composition Hierarchy System

Large system integrators may simultaneously use several sofware engineering methodologies for engineering of subsystems, e.g. functional - NASREM (Albus et al., 1987) and DFA (Elfving and Kirchhoff, 1991) - and object - (Booch, 1994; Rumbaugh, 1991; Break and Haugen, 1993; SDL, 1993; Cook and Daniels, 1994). Large object systems engineering methodologies may sacrifice the compaction of code expression gained through the use of inheritance and polymorphism in order to maintain scalability of system analysis and verification. This work proposes an outline of a methodology for integration of scalable dependable systems - FOCH - Functional Object Composition Hierarchy. The primary design goal of FOCH system analysis and design is to generate two levels of specification granularity:

1. a functional systems level granularity,
2. a functional subsystem object granularity.

Software systems without inheritance or polymorphism between objects above a certain hierarchical composition granularity are easier to analyse and verify at the systems integration level. More complicated object system possibilities such as multiple inheritance and polymorphism may be used inside object subsystems if subsystem object's interfaces mask these internal details and there is no multiple inheritance and polymorphism between subsystems. The principle system engineering goal of FOCH analysis is to achieve lexical and temporal confinement of inheritance and polymorphism to single isolated subsystems. These subsystems may then be combined using composition into larger hierarchical object systems that are easier to analyse. The FOCH methodology has the following steps:

1. Definition of functional and non-functional system requirements in three categories - mandatory, desirable, optional.
2. Preliminary broad and shallow review of design patterns and previous FOCH architectures to determine if design architecture solutions and/or code may be reused. Review of solution patterns can occur at higher than required abstraction levels, e.g. mathematical, and lower than required abstraction levels, e.g. code, as well as the architectural level of previous or close match solutions. It is also required at an early stage to forecast the impact of the information processing system on the organisational processes in which it is to be embedded in order to predict where system refinement at the validation stage is most likely.

3. Selection of an object analysis and design methodology, e.g. the one that the developer is most familiar with or a methodology tailored to a particular application domain. The selection of a methodology usually implies a restriction of CASE tool selection options.

4. Selection of a functional system analysis methodology that satisfies similar ease of usability or suitability of analysis and design language constructs for solution of the given software engineering problem.

5. Analysis top down of a system into functional subsystem components of a certain complexity granularity.

6. Initialisation of object subsystem analysis of each functional subsystem component with the subsystem results of the functional analysis. Polymorphism and multiple inheritence are prohibeted between object subsystems.

7. Analysis of the object subsystem specifications in order to determine overlap of specifications. The goal of this level of analysis is to determine a set of minimal spanning object base classes of the functional subsystems. Object subsystems may internally use multiple inheritence and polymorphic derivations of these base classes.

8. Synthesis of the object subsystem specifications leads to an analysis of how the top down functional analysis can be restructured to lead to an optimal:

   (a) containment and compact isolation of estimated object subsystem reconfiguration and maintenance hot spots. System reconfiguration and maintenance[9] hot spots must be isolated and contained in object subsytems. Reconfiguration and maintenance hot spots must be avoided at the subsystem interaction, i.e. at the systems level.

   (b) containment and compact isolation of object subsystem failure conditions and error handling. If an object subsystem fails, this failure impacts other subsystem objects. Failure containment and isolation analysis is required to identify and rank the minimum number of maximal size object subsystem faults such that an object subsystem failure has a minimal impact on other subsystems. The goal of object subsytem fault analysis is to orthogonalise object subsystem failure impacts on other subsystems in order to maximise system dependability at the specification level before introduction of object subsystem replication and persistence. The ranking of likelihood of subsystem failure is an input into subsystem performance and dependability modelling and prediction.

   (c) software structure for scalable performance modelling and dependability analysis. Hierarchical performance models of the integrated subsystems can be optimised in simulation in order to eliminate integrated system scaling bottlenecks before moving to the implementation level. There are two levels of system performance and dependability optimisation:

      i. optimisation of pure design subsystems before implementation. In this case it is not possible to generate subsystem performance model calibration data. Optimisation is defined on key design free parameters.

[9] System implementation and maintenance impact system design, including system architecture in case of radical implementation or maintenance decisions, during the system lifecycle. Hence an important software lifecycle task is to upgrade the system design and architectural specifications in line with the system implementation state in order to prevent a divergence between the system architect's view and the developers' views of the software. If the implementation architecture of a large piece of software decomposes, for instance by hacking of the base classes, the software is converted from an asset into a liability since tactical implementation decisions made by developers will at best be incompatible and at worst generate a combinatoric explosion of interaction bugs.

ii. optimisation of system integration of implemented components. It is possible to calibrate subsystems models to system operation measurement data. Optimisation specifies parameters of subsystem interactions that are optimal for a set of application configurations and application performance scenarios.

9. Synthesis of the object subsystem specifications after optimisation of subsystem hot spot containment, optimisation of subsystem fault impact on other subsystems and optimisation of subsystem simulated performance and dependability requires a secondary synthesis of the top down functional decompostion of the object subsystems.

10. Iteration of the design synthesis steps until the functional system decomposition stabalises and the object subsystems are optimal.

11. Implementation of prototype design subsystem interfaces of primary functionality and determination of mismatch between implementation and requirements. If mismatch is large, the fall back position is to renegotiate the system requirements.

12. Verification of the design prototype. This leads to finalisation of architectural synthesis.

13. Implementation of the system.

14. Verification of the system. If the system does not meet requirements, the fall back position is to return to FOCH step 8. If the system implementation does not verify performance requirements, return to step 8.(c) to fine tune the system by simulation.

15. Validation of the system.

16. Maintenance of the system.

17. Organisation and storage of design patterns and solution components for reusability.

FOCH permits a system designer to use the graphical and lexical system and architectural notations that are most convenient or efficient to represent a system in a given problem domain. System verification and validation should be orthogonalised from functional object subsystem verification and validation where possible. The goal of the FOCH methodology it so simplify system verification and validation analytical steps 12 and 14 since these steps are the primary barrier to deployment of scalable systems. To achieve this, it is necessary to make the system requirements and specification analysis and design steps more complicated.

If it is required to revise subsystem implementation details during system maintenance, to incorporate improved processing technology elements, if hot spot analysis has been successful, revision of the subsystem implementation details will have minimal functional impact at the system functional level and easy to analyse impact at the non-functional level, e.g. performance. Thus the successful application of the FOCH methodology leads to system developments that combine the advantages of object systems - flexibility of upgrade and revision of implementation details (e.g. change of implementation languages, databases etc.) - with the advantages[10] of functional analysis (comparatively easier system verification and validation).

Step 2 of the FOCH involves a review of architecture design patterns. Design patterns:

[10] Since the FOCH method is specification analysis and synthesis intensive, if system specification syntheses are not deep enough, it is possible that the FOCH method could lead to the development of a system that combines the disadvantages of functional and object system analysis and design. In this case the resulting system will be functionally brittle, it will be difficult to revise object subsystems and verification and validation will be impossible.

1. are the right level of granularity for systems architectural definition. A system architecture definition must be only as complicated as is absolutely necessary.

2. may facilitate communication of a system architecture to developers if a system architect can enumerate a list of design patterns which developers are familiar with at the computable function, algorithm and implementation level.

3. ease prediction of implementation difficulties. For complicated pattern combinations (for instance reflection plus replication plus synchronisation plus caching) developers may know that the pattern cannot be implemented cost effectively or effeciently. Developer input into the architecture specification thus avoids the inclusion of difficult to implement pattern combinations that make system verification and validation difficult.

Many software design patterns are defined by a five level hierarchy of groups. See §7.3 for a definition of groups. It is useful to define this hierarchy to make the notion of software pattern that is communicated between architects and developers precise. Let

1. let a be a software architecture pattern. a can be represented by an architecture pattern language a^{r_1}, $r_1 = 1, \ldots, z$ that spans description of patterns $a, \ldots, c$. The invariance at this level is simply that a is a pattern, i.e. that a is an invariant of the next five levels.

2. $a_i, i = 1, \ldots, y$ denote a software architecture pattern group at a functional level. a_i can be represented by different architecture design notations $a_i^{r_2}$, $r_2 = 1, \ldots, x$.

3. ${}^{j}c_{a_i}, j = 1, \ldots, q$ denote a computable function (see Definition 6.1 and Definition 6.32) that is an instance j of a design pattern a_i. A composed computable function may have different functional representations $c_{a_i}^{r_3}, r_3 = 1, \ldots, w$ determined by function group structures in the class of composed computable function. See (Davis, 1958).

4. ${}^{k}g_{{}^{j}c_{a_i}}, k = 1, \ldots, p$ denote an algorithm k to evaluate a computable function j instance of pattern a_i. An algorithm can be represented $g_{{}^{j}c_{a_i}}^{r_4}, r_4 = 1, \ldots, v$ by various algorithmic level specification languages, for instance flowchart or object interaction charts.

5. ${}^{m}l_{g_{{}^{j}c_{a_i}}}^{r_5}, m = 1, \ldots o; r_5 = 1, \ldots, u$ denote implementation language r_5 implementation m of the algorithm k evaluation of computable function j instance of pattern a_i. An implementation language only has one representation, itself.

Figure 6.8 presents the relations between hierarchical concept groups from system design to implementation levels. A group element e at level d is mapped to a group at a lower level $d+1$ by a function ${}^{e}\lceil_{d-1}^{d}$.

While an algorithm is close to the implementation level, it is useful to distinguish a group structure at the implementation language level since different developers can code different versions[11] of the same algorithm in the same implementation language. The five level hierarchy that maps design pattern to implementation groups is illustrated with a simple example software pattern - the loop. A loop is a pattern at the design patterns level. At the architectural design level, one can distinguish loops for nominal event a_1 and non-nominal event processing a_2. The loop property is an invariant of nominal or non-nominal event processing. At the computatable function level, a loop can be defined

[11] This practice occurs in fault tolerant software engineering, for instance of interplanetary probe control software, in order to minimise subsystem replica bug correlations. Different versions of the software coded by different development teams vote on subsystem outputs. The majority vote is the computed function.

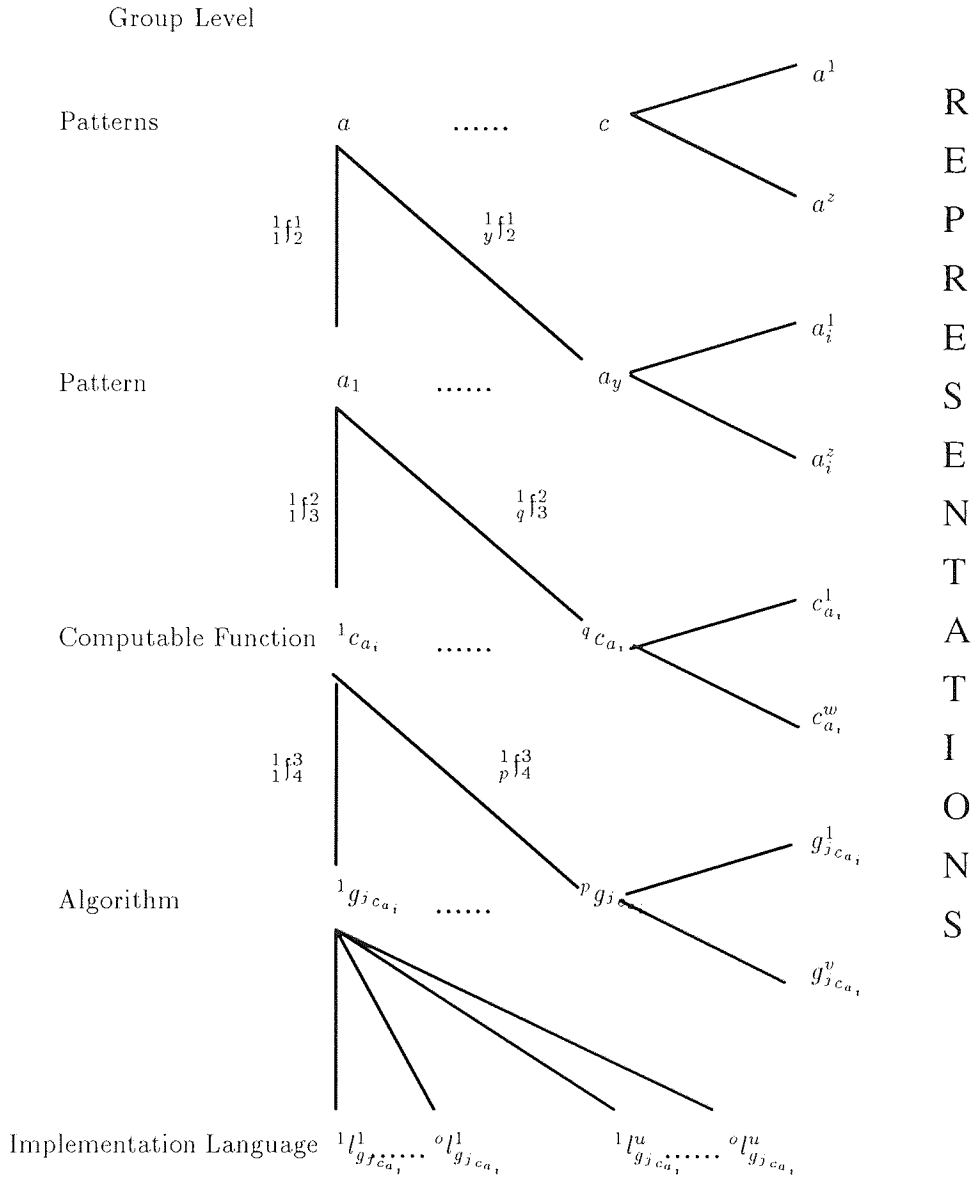

Figure 6.8: Hierarchical Concept Groups from System Design to Implementation Levels.

as a set of operations that satisfies a loop termination predicate. At the algorithm level, loops can be organised by `while`, `for` or `goto` structures. At the implementation level, these structures have different syntaxes in different languages and a developer has a great deal of freedom to decide the internal data structure and operation sequence of the loop in a particular language.

After a certain familiarisation period, communication between an architect and developers about pattern a_i subsumes the four underlying hierarchical group levels of knowledge during requirements refinement and initial functional and object system decompositions. A suitable architectural pattern notation make it easier to determine if an architecture can implemented, likely hot spots, likely fault areas and possible performance bottlenecks based on the collective developer knowledge of the details at the implementation level. It is important to collect and synthesise bottom up implementation constraint knowledge before a top down analysis in order to reduce the fragility of system analysis. Design patterns are also useful at Foch step 11 during architect/developer implementation of pro-

totype design subsystem interfaces of primary functionality and determination of mismatch between implementation and requirements. Step 11 of system development provides the developers with an opportunity to demonstrate the problems at a small scale with particular architectural decisions. The architect should develop some of the prototype or at least walk through the code to refine and concretise his design intuitions. The usual outcome of prototype implementation experience is to reduce the scope of the architecture and increase specification focus on the core system patterns and their interactions.

During FOCH analysis step 8.(a), it is necessary to perform a detailed system exception class design. Strategic system exception handling design requires a singly rooted inheritance tree of exception classes. A monolithic hierarchy of exception classes provides system robustness in all situations, i.e. all errors are caught, and the system fails only if the core memory segmentation fails and dumps. For example, if the `main()` function, in C++ object representation language, uses a `try` block that catches all possible exceptions `catch(...)`, this catch-all block logs subsystem exceptions that are not caught, and possibly restarts `main()`. A monolithic hierarchy of exception cases allows `main()`'s catch clause to extract information from the exception object by means of services provided by the root class of the exception hierarchy. This allows a more detailed description of the unknown exception to be logged. It is also necessary to use the decorator design pattern to wrap objects with their system exception handling code. Thus if A is a subsystem level object and there are three levels of exceptions - viz. g generic, c subystem specific, and s system object specific, then it is necessary to wrap A with g, c decorator objects. g, c error decorators remove subsystem infrastructure error trapping from around object invocations at the systems level. This makes code more compact and easier to understand. This makes all subsystem exception handling transparent to the application programmer at the system object calling level. System programmers are only responsible for definition of s level error trapping.

The FOCH method for iterative sequential hierarchical functional and object subsystem analysis, specification and design is designed for specification of large distributed robotics and automation systems. The above outline of a FOCH system design method requires substantial work to convert it into a mature method. In particular it is necessary to refine the method by testing it on specification and implementation of a small FOCH system before progressing to large systems analysis. FOCH also lacks an architectural language to describe the functional and object components and for instance a high level description of the object component hot spots. Generation of a FOCH architectural language is a future research avenue. A FOCH architecture language should cleanly express solutions to several system design objectives - design for dependability, design for scalability and design for extendability. To engineer high performance scalable distributed systems, FOCH design processes are structured around definition of dependable scalable system and subystem interaction performance models. See §6.5.3.

6.4 Integration of Distributed Robotics and Automation Software Systems

The software engineering complexity of the implementation of a simple prototype distributed robotics programming and control environment of §6.1 reveals why such systems are increasingly being constructed on open standards - development cost in time and resources. Large robotics and automation system vendors such as GEC, Siemens, Bosch, Mitsubishi, Sumitomo, Alcatel, and ABB have larger more expensive proprietary versions of this type of robot programming environment. Robot manip-

ulators are elements of distributed manufacturing systems. Advanced manufacturing involves coordination of hundreds of thousands of devices and computational objects. There are many problems to implement and deploy high performance distributed robotics and automation systems with soft real-time constraints of high system availability and reliability or hard fault-tolerant constraints. The ISO/ITU Open Distributed Processing reference model specification outlined in Appendix A indicates the types of functionality that are necessary to support large scale rapidly deployable and reconfigurable robotic and automation systems. The ODP specification is at quite a high level of abstraction and efficient implementation of the ODP specification at the technology level is a complicated software engineering undertaking. See Appendix B. Robotics and automation systems developers require ODP-technology level tools and library components built on international standards to easily implement and reconfigure secure reliable mixed synchronous and asynchronous transaction based multi-tier architectures with system management interfaces. Current integrated CAD/CAM/CIM object based industrial robotics and automation systems provide a substrate for advancing sensor and model based agile made-to-order single lot automated manufacturing.

Robotics and automation systems based on ODP should comply with three specifications for standardised integrated manufacturing applications:

1. the ECMA/NIST Standard for Exchange of Product (STEP) CAD data for specification of numerous aspects of large production systems, e.g. car and ship assembly by robots (Owen, 1993). The core of STEP is the EXPRESS geometry and constraint language that supports CAD/CAM data interchange across vertical manufacturing sectors. STEP system specifications are hierarchically layered down to the low level distributed object carrier technologies such as CORBA (Siegel, 1996) and DCE (see (Mullender, 1989)).

2. Profibus (Bender, 1993) and other fieldbus protocols in addition to the TCP/IP stack underlying socket communication. Profibus is the emerging standard European automation protocol.

3. ISO object oriented manufacturing standards, e.g. OSI Manufacturing Message Specification (MMS) - c.f. (Usländer, 1995b). The ISO MMS standard defines object hierarchy representations of production elements from factory subunit to tool control resolution. MMS is usually deployed in TQM ISO 9000+ JIT production systems. MMS objects are quite similar to Profibus (See §1.1, pg. 28) Fieldbus level objects, i.e. the Fieldbus message specification level 7.

4. standard distributed object manufacturing application development toolkits such as CCE. See (Usländer, 1995a).

The major automation and robotics vendors compete to develop the highest performance and heaviest functionality implementations of distributed manufacturing applications based on the combination of these standards and underlying ODP carrier technologies such as:

1. the Distributed Computing Environment (DCE). DCE is a remote procedure call (RPC) library that encapsulates call semantics and message data from underlying operating system variablity, together with programming hooks for RPC authentication etc. DCE is a relatively mature and relatively stable technology but it is unwieldy to program because its programming primitives are close to lower level systems programming syntax.

2. the Common Object Request Broker Architecture (CORBA). See Appendix B. CORBA decouples object interface definition from implementation languages. CORBA is a specification of object interface definitions and distributed object services that facilitate the construction of service based distributed object architectures.

3. Active X (i.e. the Distributed Common Object Model (DCOM) and DCE[12]). Active X enables the replacement of expensive non-real time proprietary industrial protocols with TCP/IP and cheap inefficient generic PC software. Active X is scheduled for market release in late 1997. Active X will include Common Management Information Protocol (CMIP) based system management, Viper 2 phase commit object transactions, and Falcon message store and forward. The main extension requirement for Active X in late 1998 is likely to be security and security management. The mapping from Active X to Java is also likely to be generated by late 1998 when it becomes clear which organisation controls the emerging Java standard.

4. Java is a simple object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multi-threaded and dynamic language. Java byte code executes on a Java virtual machine which is ported to different operating systems. In manufacturing markets, Java may become a good language for system integration if compilers improve. Java was developed by Sun Microsystems and it is the management strategy of this company to persuade ISO to standardise this proprietary language. Microsoft is also attempting to standardise the Java language. This Java standardisation conflict may result in a Java language and system specification bifurcation in 1998. This would severely damage Java's viability as a system integration core implementation language.

CORBA is a candidate technological language for implementation of ODP. See Appendix B. CORBA application system architectures are flat horizontal client server service systems. In principle this simplifies software architecture definition. The main function of a CORBA system architect is to design and define interface definition language specifications for services and to simulate non-functional, for instance performance, characteristics of the specified system in order to predict whether an implementation can be verified and validated. Unless a CORBA system architect uses a meta IDL specification layer, system IDL specifications may be brittle and non-reusable as bespoke low level declarative language client server API specifications. CORBA is useful for integration of valuable legacy applications written in different languages executing on different operating systems and this is its function in most applications. CORBA may become a large system integration core language but there are technical and commercial barriers to its realisation of this status.

CORBA 2.0 is a comprehensive distributed object systems specification. See Appendix B. The main theoretical problem with the CORBA specification is that certain critical parts of the specification are undefined, inconsistent or cannot be implemented. The CORBA specification is a product of design by ad hoc technical committees. The CORBA specification is the result of sequential compromise merging of competing specification committee outputs at an individual service level. Specification committees do not always consider the specification activities of other service specification committees and the lack of specification of service interactions is currently a weakness of the CORBA 2.0 specification.

The main commercial threats to CORBA as a systems integration language is the entry of Java[13] and eventually of Active[14] X, both tightly controlled proprietary technologies, into the market for distributed system integration languages. These threats may decellerate the penetration of CORBA of the enterprise application integration market, especially if Java and Active X are bundled free with operating systems. Despite CORBA's five year specification lead over Java, in the medium term to 2001, Java technology developments could substantially restrict the potential of CORBA to become a

[12] Microsoft has contracted the Open Group to port DCOM from NT4.0 to UNIX by implementing DCOM on top of DCE.

[13] Sun Microsystems technical strategy is to oppose Java and CORBA against Active X.

[14] Microsoft strategy is to oppose both Java and Active X against CORBA.

system integration core language. The Java language itself may become a canonical interface language. Direct language bindings of Java to other languages such as C++, Ada, Smalltalk, COBOL, Fortran, and C avoids the necessity for IDL. The main strategic risk to Java ascent into distributed systems integration technical hegemony is the possible bifurcation of the language specification into Microsoft and Sun variants. The Java languages has the following advantages over C++:

1. simpler syntax,
2. supports garbage collection,
3. supports multi-tasking,
4. embedded system development,
5. a virtual machine culture which insulates the developer from the software porting problem unless software efficiency is the primary priority.
6. built in security mechanisms.

See (Flanagan, 1996). Java does not support multiple inheritance or templates, except using workarounds, in order to keep the language syntax simple and encourage better object oriented design. Improvements in software reverse analysis and reverse engineering environments linked to Java compilers may eventually provide reverse software engineers with tools to accelerate re-engineering decisions during semi-automatic wrap of heritage software with Java. Once heritage applications are wrapped and linked to Java, perhaps via CORBA[15], they can be plugged into the growing world-wide Java web machine at intranet and internet level.

An important application area of Java is software distribution, bug fixing, and revision control over internet/intranet. The Sun Java management API architecture provides centralised storage of all system device MIBs and revisions. When a device needs to be managed, it sends its operating system and revision configuration to a Java management agent factory which makes the appropriate Java management agent dynamically. The agent query of the device MIB occurs within a 2 phase commit transaction. The distributed system can be managed by a single point of management application executing at any location in the system supporting a Java virtual machine.

After binding to a remote process supporting the Uniform Resource Locator (URL) protocol, Java can send UDP datagrams via a socket. Reliable threaded TCP socket communication is also supported. To succeed as a large systems integration language, Java requires a commercial grade scalable Java Remote Method Invocation (RMI) broker. The most likely technology to fullfill this function is an object request broker. The Java RMI supports transparent communications of invocations between virtual machines. The Java RMI should be extended for efficient support of secure real-time transaction based streams and notifications. This can be supported by CORBA implementations. Higher performance Java compilers combined with secure RMI transactions between Java virtual machines and efficient linkage of virtual machines to databases and Java system management are the basis for construction of a scalable Java-core ODP system. Java can perhaps be extended support construction of secure fault-tolerant manageable multi-million object mission critical enterprise applications. The development of the Java virtual machine into a JavaOS points in this direction. What is lacking is a scalable Java RMI broker.

[15] Given the amount of C++ libraries available however, a method for integration of C++ code with Java is necessary. Such links between the two languages are currently undefined. It is possible for Java to call C++ libraries via a CORBA stack although this is inefficient.

In the medium term to the year 2000, Active X is unlikely to be competitive against Java in the scalable real-time distributed object compiler market. Active X contains too much redundant desktop oriented code to fulfill this role. It is also difficult to implement CORBA specifications to provide this level of functionality because of specification inconsistencies and specification origin as a non-real time desktop application porting solution. In the long term Java performance may scale to large real-time systems by direct compilation[16] of Java to real-time microprocessors. Distributed real-time OS cores extended for CORBA invocation transport such as Iona Technologies Orbix can potentially support real-time distributed Java computations. See (Siegel, 1996). Basic functional and non-functional requirements for an effecient RMI-broker that can scale to support large scale distributed robotics and automation systems are defined in the next section.

6.5 Requirements for Distributed Object Event Transport Micro-Kernels

An open research area in distributed systems is scalable integration of relatively well understood subsystems and services. The analytical complexity of requirements analysis, specification, system design, verification and validation of larger distributed systems tends to exceed the capabilities of a single system architect. Hierarchical structuring of software development with concise demarcation and seperation of architectural and implementation roles and responsibilities is the best way to engineer large systems. Large sofware engineering projects with up to a thousand developers have several levels of system architecural management - implementation subsystem architects, information/data model architect and strategic architect. The frequent organisation of software developers into teams of less than ten can generate a restriction of the span of system architectural management, i.e. less than the usual one manager to ten managed resources, compared to other system developments. This can increase the number of management tiers required to coordinate software implementation. Large projects with up to a thousand developers where development teams are of average granularity of five resources may have three to six organisational matrix tiers (edges) to coordinate system architectural management.

The natural functional decomposition of software engineering to maintain the unity of system design and implementation is difficult to combine with three and four dimensional matrix organisational structures that are often found in software engineering companies. Matrix organisational forms for software engineering based for instance on a functional decomposition, a project decomposition and a vertical industry customer oriented decomposition make it difficult to enforce the unity of design and implementation when intermediate software architectural managment tiers map one to many from bottom to top, i.e. when tactical architects and implementation team leaders are responsible to managers coordinating the functional, product, and customer management decompositions of a matrix organisation. Matrix organisation is required to generate a system requirements analysis which responds to various inputs - what is technically possible, what is in demand in different markets and how what is technically possible can optimally drive market demands. Matrix organisation also facilitates TQM JIT software production and distribution to the worldwide geographic market segments. Large software engineering projects require an explicit transparent specification of the chain of strategic architectural design and decision making to tactical technical implementation decision making in order to map a unity of system design to a unity of system implementation through the matrix organisation generation of system requirement priorities.

[16] Java was originally developed as an embedded real-time language.

Matrix organisations require a clear policy for resolution of lower level decision conflicts that impact the architecture in order to preserve system architectural unity. It is necessary to overlay a decision making priority hierarchy over the three matrix management decompositions and procedures in order to verify across decompositions the consistency of requirements decision making and implementation operations management made in the three management decompositions. Without hierarchical management decision and operational conflict priority resolution procedures, verification of system architectural decision sequence integratity and operational implementation workflow optimisation is impossible. This activates the principle risk of the matrix organisational form - the decomposition of the unity of architectural decision making and the degeneration of software engineering into competing fiefdoms that rip the architecture apart. The FOCH methodology can also be applied to analysis and design of software resource organisation in addition to organisation of software development itself.

If a good software engineering organisation can be constructed, it can be applied to solution of very large distributed systems engineering problems. This section defines elementary functional and non-functional requirements for a virtual distributed kernel event transport system that will scale to support a Java-ODP robotics and automation superstructure outlined in §6.6. These requirements are generic for any scalable ODP technology language and in particular for a scalable RMI transport kernel or broker that supports Java RMI. The non-functional requirements specification is at an architectural level. The purpose of the requirements analysis is to attempt to refine some of the analytical elements of the FOCH methodology design steps 8.3 and 9 related to system design for dependability and high performance. The requirements are structured as a set of definitions.

6.4 Definition (An Object)

An object o_m is a functional encapsulated persistent service. o_m is an encapsulated state, e.g. a set of attributes, of a computational entity. o_m state changes may occur as a result of local methods or operations or via communication with remote objects via $\texttt{send}({}^{m}_{n}\mathbf{e}_t)$ and $\texttt{receive}({}^{m}_{n}\mathbf{e}_t)$ primitives of event vector $\mathbf{e}_t$ at time t from o_m to o_n. ODP requires that state change caused by transport event communication with remote objects o_n be transparent with respect to the communication mechanisms. To implement transparency, an object's identity should not depend on its location in a distributed system.

6.5 Definition (An Object Context)

An event transport core supports sophisticated context binding to objects. For instance, when a bind to a service is requested by an object, this would imply bind to the most local service satisfying default quality of service parameters. See §6.5.1, pg.140.

6.6 Definition (A Prioritised Virtual Communication Vector Event)

${}^{m}_{n}\mathbf{e}^{p}_{t}$ an event vector sent at time t from o_m to o_n with priority p is a marshalled event that causes an internal state change of o_n if it is delivered. Communication is achieved through $\texttt{send}({}^{m}_{n}\mathbf{e}^{p}_{t})$ and $\texttt{receive}({}^{m}_{n}\mathbf{e}^{p}_{t})$ primitives of event vector ${}^{m}_{n}\mathbf{e}^{p}_{t}$ at time t from o_m to o_n. Event priorities p are constraints input into event transport scheduling algorithms. Events ${}^{m}_{n}\mathbf{e}^{p}_{t} = \{{}^{m}_{n}i^{p}_{t}, {}^{m}_{n}s^{p}_{t}, {}^{m}_{n}n^{p}_{t}\}$ where:

1. ${}^{m}_{n}i^{p}_{t}$ is a synchronous invocation from o_m to o_n at time t,
2. ${}^{m}_{n}s^{p}_{t}$ is a stream from o_m to o_n at time t - conceptually a continuous invocation but technologically implemented by steam fragment transmission with error checking.
3. ${}^{m}_{n}n^{p}_{t}$ is a notification from o_m to o_n at time t, e.g. warnings, alerts, updates.

Prioritised events can:

1. be nominal - i.e. results of invocations, operations, streams,

2. be non-nominal - i.e. warnings, alerts, updates:

   (a) nominal synchronous event exceptions.

   (b) asynchronous events - push and push.

Events can also have a default context c and an associated processing exception vector ex. Thus a full event notation is ${}^{m}_{n}\mathbf{e}^{(p,c,ex)}_{t}$. ${}^{m}_{n}\mathbf{e}^{(p,c,ex)}_{t}$ is virtual if it is transported by a system of micro-kernels providing an invariant virtual interface across operating systems, e.g. UNIX, NT.

A coordinate system for objects is required to support transparent communication of events. When an object o_m binds a reference to a remote object o_n, the distributed pointer $\mathtt{ref}_{(i,n)} = f(k_i, o_n)$ contains coordinates of the local kernel[17] to which the referent is bound and the object, i.e. (k_i, o_n), for kernel k_i and object o_n. Each time an event ${}^{m}_{n}\mathbf{e}^{p}_{t}$ with priority p is called on $\mathtt{ref}_{(i,n)}$ by o_m, it is the function of the micro-kernel routing system to transport the marshalled calling event vector ${}^{m}_{n}\mathbf{e}^{p}_{t}$ to (k_i, o_n).

6.7 Definition (An Object Server)
v^{j}_{i} is a computational process server j composed of a group of objects o_m attached to kernel k_i.

6.8 Definition (A Routing Kernel Object)
A routing kernel k_i is an object to which are connected sets of objects. The function of k_i is to transport marshalled ${}^{m}_{n}\mathbf{e}^{p}_{t}$ from o_m to o_n a single object or an object group. It is not the function of the routing kernel to marshal and un-marshal events. Kernels k_i handle stream, invocation and notification traffic. It may not be practical to support stream communications if high bandwidth channels, e.g. Asynchronous Transfer Mode (ATM) networks or Integrated Standard Digital Networks (ISDN), are not available or there are high invocation and notification system traffic loads. In the case of notifications, event reception groups may be defined $(k_i, o_n) := \{(k_i, o_n)\}$. Kernels k_l may require a facility to store sequential queues of ${}^{m}_{n}\mathbf{e}^{p}_{t}$ for queuing event vectors that can be pulled by o_n. ${}^{m}_{n}\mathbf{e}^{p}_{t}$ that are pushed by o_m asynchronously interrupt o_n if o_n have registered with the kernel system $\{k_i\}$ to receive notifications of type ${}^{m}_{n}\mathbf{e}^{p}_{t}$ from o_m.

6.9 Definition (A Routing Topology)
A routing topology is a connection graph $\{T_a\}$ between k_i.

6.10 Definition (A Hierarchical Routing Topology)
A hierarchical routing topology is a connection graph $\{\{T_a\}, \ldots, \{T_z\}\}$ between k_i organised in communication tiers.

It is the function of kernel event routing algorithms, e.g. adaptive dynamic programming, to determine the shortest or fastest path to transport ${}^{m}_{n}\mathbf{e}^{p}_{t}$ from k_l to k_i. When o_n has finished processing ${}^{m}_{n}\mathbf{e}^{p}_{t}$, o_n marshals the response ${}^{n}_{m}\mathbf{e}^{p}_{t+a}$ and delivers it to k_i for delivery to (k_l, o_m). It is the function of $k_i, \ldots, k_l$ scheduling algorithms to determine the shortest or fastest path from k_i to k_l. Determination of optimal routing algorithms is a complex task, especially optimal routing of events between loosely coupled asynchronous computations. See (Bertsekas and Tsitsiklis, 1989). There are several possible routing topologies for the kernels:

1. bi-directional k_i peer to k_j peer:

   (a) static k_i leaf to higher tier k_j,

[17] Thus while the remote object reference is transparent, i.e. location invariant, the contents of the referent, the system address, are not transparent.

(b) higher tier k_j peer to peer,

(c) higher tier k_j to yet higher tier kernel k_l ....

2. bi-directional inter-kernel broadcasts to support notifications,

3. arbitrary leaf k_i peer to peer for stream communications. Stream control may occur via the normal hierarchical routing kernel system,

4. arbitrary leaf k_i peer to peer for invocation communications.

6.11 Definition (Global Event Routing and Object Load Optimisation Algorithms)
Global event routing and object load optimisation algorithms $G = \{g_i\}$ can be (Bertsekas and Tsitsiklis, 1989):

1. static:

   (a) sub-optimal, optimal - heuristic and approximate:

      i. enumerative, graph theory, mathematical programming and queuing theory,

2. dynamic

   (a) physically non-distributed,

   (b) physically distributed:

      i. non-cooperative,

      ii. cooperative - optimal sub-optimal - approximate and heuristic,

There are two inter-dependent levels of optimisation of routing in a distributed object system

6.12 Definition (The Optimal Routing of Events over Kernels)
Let routing of transport of events ${}^m_n\mathrm{e}^p_t$ by k_i be denoted $T_a|^m_n\mathrm{e}^p_t$. Let f be a function that measures goodness, e.g. minimal time, of event transport scheduling algorithms g for event ${}^m_n\mathrm{e}^p_t$ over $k_i, i = 1, \ldots, z$, where z is the total number of routing kernels. Let $g_i(k_i)$ denote scheduling algorithm at k_i, usually $g_i = g_j, i, j = 1, \ldots, z$. Then the optimal routing of events over kernels is defined by

$$T_a^{\max} = \min_{i,m,n,p,t}(\sum_t \sum_p \sum_m \sum_n \sum_i f({}^m_n\mathrm{e}^p_t, g_i(k_i))). \tag{6.4}$$

Adaptive local scheduling algorithms with best-try semantics are useful at this level. See (Bertsekas and Tsitsiklis, 1989).

6.13 Definition (The Optimal Application Load Balance)
Let application j load scheduling, i.e. assignment of o_m to k_i, be denoted A_j. Let $\bar{f}^m_i$ be a measure of average event frequency in and out of o_m linked to kernel k_i. Let $\bar{l}^m_i$ be a measure of average processing load for each event in or out of o_m attached to kernel k_i. The average load[18] on k_i is $\sum_{m=1}^n \bar{f}^m_i \bar{l}^m_i$ where

[18] The actual implementation of object load computation is an implementation level detail which impacts optimisation of kernel design so implementation details interact with the abstract mathematical level. In order to keep the kernels light, implementation of average load computations should be a management service, i.e. performance evaluation function, that attaches to a kernel only when the routing optimiser is optimising. If an application stabalises, and a global optimal loading is reconfigured, the joint optima of loading and event routing is sub-optimal if management processes are in place for measuring average load. To reach final performance optima, management measurement probes of load imbalance should detach from kernels and servers to leave a pure unmanaged optimal application if the prior is that this is likely to remain a stable processing load optimum.

n objects o_m are attached to k_i. Given a specific application j, optimisation of application object load distribution, i.e. optimisation of $\{k_i, o_m\}$ is defined by

$$A_j^{\max} = \min_{i,j}(\sum_i \sum_j ||\min_b(\sum_{m=1}^{b} \bar{f}_i^m \bar{l}_i^m) - \min_c(\sum_{n=1}^{c} \bar{f}_j^n \bar{l}_j^n)||). \tag{6.5}$$

(6.5) defines the optimally balanced load as that load which simultaneously minimises the average load over all kernels by attaching subsets of o_m to k_i optimally such that all application j's o_m are attached to some k_i.

6.14 Definition (Coupled Optimisation of Global Event Routing and Object Loading) Event transport global optimisation is application j dependent. Interaction of event routing optimisation and object loading optimisation is a complicated non-linear optimisation problem since a lightly application loaded kernel, i.e. k_i with few o_m attached may for instance be a bottleneck in inter-kernel event routing. Combining (6.4) and (6.5) to define the global optimal application loading and event routing configuration leads to a coupled minimisation problem to determine

$$\begin{aligned} TA_{a,j}^{max} = \alpha[\min_{i,j}(\sum_i \sum_j ||\min_b(\sum_{m=1}^{b} \bar{f}_i^m \bar{l}_i^m) - \min_c(\sum_{n=1}^{c} \bar{f}_j^n \bar{l}_j^n||)] \\ +\beta[\min_{i,m,n,p,t}(\sum_t \sum_p \sum_m \sum_n \sum_i f({}_n^m \mathbf{e}_t^p, g_i(k_i)))]. \end{aligned} \tag{6.6}$$

where α, β are event routing vs application loading optimisation Lagrange multipliers. Metric scaling of (6.6) is required since the units of optimisation of routing are generally seconds and measurements of load balance are generally at the minute level. Once units are normalised for coupled optimisation using for instance BFGS if the coupled optimisation function can be continuously approximated, then sequential steepest descent alternatively on routing and loading is a simple optimisation strategy. Many advanced numerical optimisation heuristics may be deployed to search for global optima of good sub-optima. See (Bertsekas and Tsitsiklis, 1989; Press et al., 1992). There are several classes of event routing and object loading optimisation algorithms that may implement (6.6) for a given set of applications j executing on a distributed computing mesh.

There are several problems to be solved at an ODP technology level to support event routing optimisation and application load balancing optimisation. For instance, it is necessary to solve the remote pointer object migration problem to implement load optimisation. Standard reference counting garbage collection techniques can to solve this problem. An object o_m has a remote pointer $\mathtt{ref}_{(i,n)}$ to o_n. If o_n detaches from k_i and attaches to k_z as a result of a load optimisation computation, it is necessary to send a o_n migration notification event to the set of objects $\{o_m\}$ that call o_n. In order to do this, it is necessary that o_n maintain a list of the objects $\{o_m\}$ containing $\mathtt{ref}_{(i,n)}$ in order to send a migration update event ${}_m^n\mathbf{e}_t^p \equiv \mathtt{ref}_{(i,n)} \rightarrow \mathtt{ref}_{(i,z)}$. Implementation of a remote pointer reference list $\mathtt{ref}_{(i,n)}$ can also be used by a memory garbage collection algorithm.

An alternative to a notification broadcast of distributed pointer reference updates if a referenced object migrates across kernels or nodes is use a relocation service for a domain. A domain is a set of kernels and nodes with federations of relocation services connected optimally in a similar manner to optimisation of object loading. If o_m sends ${}_n^m\mathbf{e}_t^p$ to o_n based on $\mathtt{ref}_{(i,n)}$ and k_i detects that o_n is no longer attached to k_i, then k_i may query the relocator service RS_{D_d} for its domain D_d to find an up-to-date reference for o_n. If RS_{D_d} locates o_n attached to kernel k_j, it returns an updated reference $\mathtt{ref}_{(j,n)}$ to k_i which can do two things:

1. route ${}^{m}_{n}\mathbf{e}^{p}_{t}$ to k_j for processing and return the result ${}^{n}_{m}\mathbf{e}^{p}_{t+1}$ to o_m transparently, followed by notification of o_m to update $\texttt{ref}_{(i,n)}$ to $\texttt{ref}_{(j,n)}$.

2. forward $\texttt{ref}_{(j,n)}$ to o_m with new event ${}^{m}_{n}\mathbf{e}^{p}_{t+1}$ that $\texttt{ref}_{(i,n)}$ should be updated by an ${}^{m}_{n}\mathbf{e}^{p}_{t}$ system exception. It is the function of o_m to update $\texttt{ref}_{(i,n)}$ to $\texttt{ref}_{(j,n)}$ and resend ${}^{m}_{n}\mathbf{e}^{p}_{t}$ based on $\texttt{ref}_{(j,n)}$.

If RS_{D_d} fails to locate o_m, it

1. queries $RS_{D_e}, \forall e \neq d$, all other locator services, i.e. it queries the locator federation. If this fails, then go to the next step.

2. or returns a system exception to k_i which relays an exception notification event to o_m.

A transparent kernel based query of the locator service is preferable from the programming point of view to keep the global system exception hierarchy simple and efficient at the application development level.

6.15 Definition (Heuristics for Elimination of Kernel Event Processing Bottlenecks)
While a global optimisation correction of event routing load drift and kernel load imbalance across communication mesh topology is optimal, it is non-trivial to engineer this type of system correctly. Several heuristics are useful to assist a global predicate computation to minimise kernel and server migration triggered by relatively transient load spikes:

1. Kernel auto-determination of kernel switching priority in the OS process scheduler can remove short transient event queues if real-time critical application elements are not connected to the kernel superstructure.

2. temporary switching from priority based event processing and routing to local least loaded kernel routing. This is necessary to remove staggered queue bottlenecks between sequentially overloaded kernels.

The basic function of the micro-kernels is to transport communication events, i.e. invocations, streams and notifications. It is not the function of the core to marshal, un-marshal and define object bindings to different languages. Let

1. M denote a marshal operation and M^{-1} denote an un-marshal operation. Marshalling converts object represents to vector representations for transmission by vector packet oriented communication protocols.

2. E denote an encrypt and E^{-1} a decrypt operation. This is a service between k_i and o_m, a dedicated security micro-kernel. See §6.5.1.

3. T denote transportation of an event over a protocol P_d from o_m to o_n and T^{-1} transport of an event from o_n to o_m. This occurs at the kernel level.

4. C denote a prioritised concurrency service for operations on o_n. This is a service between a capsule containing o_m and o_m.

5. P denote processing of object o_n of event ${}^{m}_{n}\mathbf{e}^{p}_{t}$. This occurs within object o_n.

The basic remote invocation loop involves a state change of o_m, $s_{o_m} \rightarrow s'_{o_m}$ where

$$s'_{o_m} = M^{-1}(E^{-1}(T^{-1}(E(M({}^{n}_{m}\mathbf{e}^{p}_{t+1} = P(C(T(E(M(({}^{m}_{n}\mathbf{e}^{p}_{t}))))))))))). \tag{6.7}$$

Distributed system state dynamics are determined by iteration of (6.7) over all o_m, as $t \rightarrow t+1$.

6.16 Definition (The Master Kernel Group)

The master kernel group controls system integrity. The master kernel group processes kernel system state information events, not application events. The master kernel group has two functions - system integrity control (SICO) and micro-kernel global loading and routing optimisation. The primary SICO functions of the master kernel group are:

1. reinitialisation of the kernel system to restore system sanity after catastrophic fault showers. The master kernel group has system integrity control which overrides master kernel group infrastructure optimisation which overrides local kernel decision making. Override is necessary to suppress operational processing of fault ridden infrastrure after a catastrophic fault event shower. To implement this it is necessary for the master kernel group executive to periodically transfer control to the SICO to test the reboot predicate override of faulty executive and fragmenting kernel infrastrures and or network systems.

2. In this case, SICO reboots the kernel system to a previously sane state if this is known, via state logging.

3. A kernel SICO subroutine may notify network SICOs and computer SICOs if repeated restoration of kernel system to sanity is not stable because of network and computer faults. In this case, the network and computer SICOs of a system management platform should be able to remote reboot each system element. It is required that the master kernel group SICO is guaranteed sane by hardware replication up to fail safe level five depending on hard real-time system requirements.

4. the master kernel group should be capable of riding over transient faults, e.g. a single kernel failure of a kernel replica group. The master kernel group should also signal human application controllers of immanent system failure or non-transient fault events at the kernel and application system.

5. implementation of a policy for optimal suppression of fault shower event display to minimse human factor decision errors caused by operator panic and intervention to reset the system in the case of transient fault showers.

The communication protocol between the master kernel group and the micro-kernel infrastructure is usually one of query of kernel integrity and load, and command of micro-kernel migration and reconfiguration. Micro-kernels generally reply to queries and commands with status responses. Desing of the protocol for synchronisation of the master kernel group replicas requires attention to synchronisation optimisation to avoid the master kernel group becoming a system bottleneck. As a system scales, the synchronisation traffic of the master kernel group representations of global system state may scale exponentially depending on the synchronisation protocol.

Loading of the master kernel group is a network location optimisation problem. Master kernels should be loaded such that each master kernel is responsible for a maximal micro-kernel subset such that summed communication distance, weighted by kernel processing load, from the subset to the master kernel is minimal. Master kernels should also be loaded on maximally failure independent subsystems. This aspect of master kernel loading is a design input to design step 8.(b) of FOCH. Inter-master kernel synchronisation communications should not occur via the micro-kernel infrastrure in order not to compromise the SICO function. The master kernel group should communicate possibly peer to all peers via low level bespoke protocols operating on sockets with a special purpose secure communications protocol, e.g. RSA based daily interchange of DES keys. The SICO and loading optimisation functions of the master kernel group depend on the micro-kernel infrasture. If infrasture fragments, initially global system loading and routing optimisation is compromised, and finally SICO

commands to the micro-kernels fail. A particularly vulnerable sytem component for connection of the master kernel group to the micro-kernel ifrasturture are relocator or naming services. These must have higher processing robustness than communication micro-kernel elements. A master kernel infracture provides a basis for engineering of a dependable scalable Java Remote Method Invocation (RMI) micro-kernel infrastructure.

6.5.1 System Management and Security Interfaces and Meta-Transport Protocol

Design of the micro-kernel primary event processing loop should be independent of underlying invocation, stream and event transport protocols P_d in order support easy extension to other protocols. It is necessary to determine a minimum spanning API hierarchy into the most important transport protocols. A meta-transport interface design offers the most flexible foundation for development of virtual event transport micro-kernels. There are many different network communication protocols specialised to support different application requirements. Protocols range from connection-oriented to connection-less, synchronous to asynchronous, reliable to unreliable, stream-oriented to message oriented. Network protocols can be embedded in an OS kernel, e.g. UNIX System V , or decoupled from a micro-kernel and provided as a service, e.g. Mach. See (Mullender, 1989). Transport protocols are graphs connecting protocol data format objects and session transport dynamic control objects. To define a protocol interface, in general it is required to assume no a-priori knowledge of protocols in order to forward engineer a kernel mapping to evolving and novel protocols. To bind to all protocols, it is necessary to specify a meta-protocol interface for instance the x-kernel (Hutchinson and Peterson, 1991). For the generic case of arbitrary meta-protocol inter-operability, the x-kernel defines a single dynamic de-multiplexing scheme as follows:

1. Each protocol is identified by a unique system-wide 32 bit identifier.
2. Each protocol's header begins with its unique identifier.
3. Each protocol inspects the next protocol's header to get its id, and the de-multiplexes to this id.
4. All protocols without headers (i.e. virtual or optimisation pass through protocol objects) have the protocol id of the protocol that opened them.
5. A special protocol is added to all messages as they cross the user/kernel boundary.
6. Protocols that fragment messages should duplicate the high level protocol's identifier at the start of each non-initial fragment.

A protocol should also provide a function to return a formatted version of its header. The x-kernel meta-protocol covers the core functions found in almost all transport protocols:

1. request/reply signaling,
2. message encryption,
3. message encoding.
4. message fragment/reassembly,
5. message routing,

6. sliding message window,

7. transmission or reception stop-and-wait.

This set of transport protocol building blocks can be used for construction of arbitrary protocol graphs. These network transport graphs interface to the event processing kernel interface at event protocol and session object levels. For the purposes of specifying the virtual micro-kernel it therefore seems sufficient to only specify kernel event protocol and session objects. It is a complex implementation issue to determine how to link these protocol and session objects to a particular transport layer, e.g. TCP/IP. In general, connection of actual transport protocols to actual operating systems is achieved in different ways by:

1. monolithic kernel, e.g. UNIX with embedded transport protocols,

2. micro-kernel, e.g. Mach or NT, with server transport protocols.

There is a scaling design tension. To make communication efficient, i.e. to satisfy the primary design goal of system scalability, it may be required to bind a particular communication protocol tightly to a kernel. However to facilitate the virtual port of the kernel to many target protocols it is necessary to define a meta transport protocol interface. There are thus two implementation cases:

1. implementation instantiations of the meta protocol that are under the meta protocol stack,

2. in special cases, the meta protocol can be replaced by a direct bind of the micro-kernel to a given network transport protocol stack.

For optimal scaling, it is necessary to bind the combined event processing/transport kernel tightly to a monolithic or micro-kernel OS. In this case, the event processing/transport kernel should be bound to the transport device before the normal Operating System transport protocol unstack.

Micro-kernel system management requires

6.17 Definition (Kernel State Monitoring by Kernel Neighbours and the Master Kernel Group) It is necessary to design a performance measurement Management Information Base (MIB) to permit a human application system controller to configure kernels via an system management platform or via the global kernel master group. The best way to design kernel monitors is to decouple the system management state monitor class system into:

1. instrumentation probes,

2. probe selectors,

3. probe processors,

4. probe recorders,

5. probe interpreter and analyser.

Probes are passive monitors. Extensive system management of faults, configuration, accountancy, performance and security should be supported by the kernel. The core performance monitor should efficiently span parameterised collection of:

1. event processing duration,

2. event throughput,

3. kernel utilisation rate,

4. kernel event queue length,

for object loading optimisation (see Definition 6.14) and performance model calibration (see §6.5.3).

These four elements form the basic elements of the Management Information Base (MIB) of a single kernel. There are two means for an external management platform to access the micro-kernel MIB

1. via agents and standard management protocols, e.g. Simple Network Management Protocol or Common Management Information Protocol,

2. via the kernel system itself.

The design of the kernel should incorporate transmission of MIB information via the kernel system to a standard management platform and the master kernel group in order to minimise extra-kernel communications in the system.

Distributed system security has the following dimensions:

1. data confidentiality,

2. data integrity,

3. identification and authentication,

4. access control,

5. auditing,

6. communication security,

7. security administration,

8. non-repudiation,

9. assurance.

The interaction of these security aspects is difficult to analyse. It is difficult to validate implementations due to for instance the distortion of security by subtle system operation timing variations in addition to the usual bug quality control problems. See (Mullender, 1989). One of most interesting distributed system security designs being prototyped is the National Security Agency (NSA) Synergy portable micro-kernel based security architecture. Synergy is designed to solve the hard multi-level secure (MLS) heterogeneous systems problem. The Synergy design is based on separation of servers for:

1. security - i.e. access control. The security server control access, but the microkernels enforce access policy,

2. audit - centralised state log,

3. cryptograpy - centralised mechanisms and algorithms, based for instance on a RSA-250 key,

4. network server - x-kernel protocol provides secure multi-level protocol communication,

5. authentication server. This provides user authentication and system authentication,

6. O/S server.

The Synergy invocation security clearance sequence involves access to the authentication server, followed by access to the security server with complete system log to the audit server. Authentication, security, and syslog servers can be called by a micro-kernel. The most complicated element of Synergy which could be factored into a micro-kernel security system is the combination of x-kernel and the crypto-server. It is preferable for performance scaling reasons to absorb the authentication and security function into the kernel as much as possible. The absorbed authentication and security functions could communicate inter-kernel and with the x-kernel and Crypto server via the Generic Security Service Application Programming Interface (GSSAPI). See (Mowbray and Zahavi, 1995).

The Synergy crypto-server is based on the RSA algorithm. System security is particularly difficult to implement correctly. Each code bug is a security bug that can be exploited by skilled developers. Bug ridden code cannot be made secure. An apparently secure system is a greater security threat than an insecure system since an insecure system is known to be insecure while an apparently secure system is not known to be insecure. Security is a core functional design issue which impacts performance. The distributed Kerberos authentication service for instance is not scalable so incorporation of Kerberos based authentication in a micro-kernel system will result in system non-scalability.

A micro-kernel should also offer a Quality of Service (QoS) interface to the application superstructure and other micro-kernels as part of the management functionality if this is integrated into the kernel system itself. If applications wish to use the QoS interface to the core, it is necessary for applications to query the MIB of the local kernel element. Application query of a kernel component of the core MIB is non-updating. Other QoS parameters for instance for real-time, are aggregated by a hierarchical management system. There are two levels of application QoS which a core can support:

1. global ATM style. This can be approximated by collection of core MIB statistics and auxiliary service statistics by the management system. QoS is probabilistic by nature and Dirac QoS on bounded invocation latency is only possible in a low throughput regime.

2. per invocation. This is an interesting theoretical problem. In order to determine QoS per invocation, it is necessary to normalise the statistics of the local transport core by statistics of the transport route. Per invocation introduces a level of QoS approximation recursion into the system based on the response of the client to the QoS query. If QoS is within some bandwidth of parameters, the client will invoke and increase the system load, i.e. reduce the QoS bandwidth at time $t + 1$. However this can be predicted at time t. This is thus an input into advanced predictive load balancing to flatten the combined QoS parameters for local sections of transport core.

In the case of dynamic load balancing, see (Bertsekas and Tsitsiklis, 1989), by object migration, to maximise QoS matrix globally or sub-matrices for local servers, it is necessary to flag objects to the system which explicily may require a location dependence, for instance to connect to local devices. Object migration should be supported in general by OS process migration primitives. In the case of migration of a server object within a server with several server objects, it is necessary to fork the process and migrate the object sub-process.

A micro-kernel implementation should be easy to program, debug, maintain and it should facilitate extension of the system, i.e. have a compact efficient API supporting maximum distribution transparency. Since any micro-kernel system will eventually be superseded by a superior distributed systems integration technology, it is necessary to design in hooks to facilitate future system integration of the kernel infrastructure when it attains legacy status.

The following superficial enumeration of non-functional requirements indicates that engineering a scalable dependable high performance RMI-broker is a difficult software engineering problem. System dependability is generally inversely related to system scale.

6.5.2 Micro-Kernel System Dependability and Scalability Requirements

The basic kernel processing model is one of an stimulus-response. Application events provide stimulus to the kernel infrastructure which responds to these stimuli by stimulating remote applications and possibly delivering the response of these remote applications to the original stimulator application. The most important non-functional requirement for the event transport micro-kernels is dependable scalability. System dependability and scalability are both related to system fault probability.

6.18 Definition (System Fault Probability)
System fault probability, F_s^p, is the probability of a fault at any system level. If a indexes a subsystem, ${}^aF_s^p$ is the fault probability of subsystem a. For analytical purposes, subsystem failure probabilities can be assumed[19] to be approximately independent. In this case,

$$F_s^p = \prod_{a=b}^{c} {}^aF_s^p. \tag{6.8}$$

(6.8) implies that system fault probabilities in general are a function of system scale where scale is defined by c, the number of subsystems. A more complex measure of scale is required for software engineering purposes. Let:

1. c be the number of processing nodes, usually uni-processors, in a system,
2. h be the number of hardware interconnections between nodes,
3. $\bar{v}_i$ be the average number of server processes per kernel k_i,
4. $o_m^{v_k^j}$ be the average number of server objects o_m in server v^j attached to kernel k_i,
5. $\bar{s}_j$ be an average server object request load, measured in CPU cycles and memory usage, for an average application j,
6. $\bar{r}_j$ be an average number of requests into and out of an average server for a given object,
7. $\bar{r}_s$ be the average size of requests transported over the network.

6.19 Definition (System Scale)
Let Γ_s be a measure of a distributed system scale

$$\Gamma_s = f(c, h, \bar{v}_i, o_m^{v_k^j}, \bar{s}_j, \bar{r}_j, \bar{r}_s), \tag{6.9}$$

to a first order approximation

$$\Gamma_s = c \times \bar{v}_i \times o_m^{v_k^j} \times \bar{s}_j \times \bar{r}_j + \frac{c \times \bar{v}_i \times \bar{r}_s \times \bar{r}_s}{h}. \tag{6.10}$$

[19] This approximation assumption must be corrected close to the system deployment phase since F_s^p is greater than the product of subsystem failure probabilities because subsystem failure probabilities are usually not independent, even if there are no software engineering flaws. Failures are correlated. A failure of a router may cause a failure of a capsule which may cause a failure of a server.

With heterogeneous distributed systems, as an application system scales, i.e. Γ_s increases, the probability of system failure F_s^p increases. $F_s^p = f(\Gamma_s) \approx 1$ for $\Gamma_s > \Gamma_s^{\max}$. For large Γ_s, component, failure is likely at any level, e.g. network, router, OS, kernel, server or server object. System scalability requires system fault tolerance to faults at any and all of these levels. Asymptotically, $F_s^p \rightarrow 1$ as $c \rightarrow \infty$ and ${}^aF_s^p \rightarrow 0$. Minimisation of a subsystem a fault probability ${}^aF_s^p$ increases the upper bound c at which system fault probability is unity. A large scale system with unity fault probability must be highly dependable.

System dependability D_s is a combination of system reliability, availability, safety, performability and maintainability.

6.20 Definition (System Availability)

Availability, A_s, is the probability that a system is operating correctly and is available to perform its function at time instant t.

6.21 Definition (System Reliability)

Reliability, R_s, is the probability of correct system function over a time quantum.

R_s is an integral function of A_s. Low availability reduces reliability.

6.22 Definition (System Maintainablity)

System maintainability M_s is the probability that a failed system will be restored to an operational state within a specified time quantum. Restoration includes problem detection, location, physical repair, and system reinitialisation to sane operation after failure.

6.23 Definition (System Safety)

S_s, system safety is the probability that a system performs correctly or degrades safely, i.e. does not result in human death or injury. If a system does not operate correctly, it should shut down, i.e. fail, safely. Safety is the probability that all safe actions succeed. Safety is a stronger constraint than reliability since a reliable system may perform its functions correctly and still kill people through a design error for instance. $S_s = f_1(R_s, A_s, P_s, M_s) + f_2(sd|f_x)$ where $sd|f_x$ represents system shut down sequence for un-recoverable fault f_x.

R_s, A_s, P_s, M_s, S_s are probabilistic functions of system element fault probability functions. R_s, A_s, P_s, M_s, S_s are system level function probability measures. System dependability $D_s = f(R_s, A_s, P_s, M_s, S_s)$.

6.24 Definition (System Failure Rate)

F_s, the system failure rate is the expected number of system failures over a given temporal quantum. Failure rates usually have three phases:

1. F_s^a system burn in failure rate,
2. F_s^b stable operational failure rate,
3. F_s^c system burn out failure rate.

At each stage of a system lifecycle - burn in, operational, and burn out - $A_s = F_s^a, A_s = F_s^b, A_s = F_s^c$. The failure rate function $F_s(t)$ is related to the reliability function $R_s(t)$ by

$$\frac{dR_s(t)}{dt} = -F_s(t)R_s(t). \tag{6.11}$$

The rate of change of reliability is the product of the current reliability by the failure rate. To determine the change in reliability, it is necessary to integrate (6.11), a stochastic (in $F_s(t)$) differential equation, after assuming a form for $F_s(t)$, e.g. Weibull,

$$F_s(t) = \alpha\lambda(\lambda t)^{\alpha-1}. \tag{6.12}$$

In the simplest case of a steady failure rate, i.e. $\alpha = 1$ and $F_s(t) = \lambda$. If $\alpha > 1, F_s(t)$ increases. If $\alpha < 1, F_s(t)$ decreases as t increases. The integral of (6.11) is

$$R_s(t) = e^{(-(\lambda t)^{\alpha})}. \tag{6.13}$$

6.25 Definition (System Fault Coverage)
Fault coverage, C_s, is the conditional probability that given the existence of a fault, the system recovers. System recovery requires fault detection and trapping. It is hard to compute fault coverage since it is necessary to integrate over all fault types that can be located, contained and recovered. The fault detection coverage factor is the number of faults detected divided by the total number of faults (assumed measurable in principle). $C_s = f(F_s)$.

6.26 Definition (The Mean Time to System Failure)
$\bar{T}_s^F$, the mean time to failure of system s is $\bar{T}_s^F = \int_0^\infty R_s(t)dt$. If $R_s(t)$ is exponential, e.g. 6.13, with mean failure rate λ, $\bar{T}_s^F = \frac{1}{\lambda}$.

6.27 Definition (The Mean Time to System Repair)
After a system failure, $\bar{T}_s^U$ is the mean time to repair system s to bring it back up. $\bar{T}_s^U$ is difficult to determine. $\bar{T}_s^U$ is often tested by deliberate fault injection schedules. If μ is the repair rate, $\bar{T}_s^U = \frac{1}{\mu}$.

6.28 Definition (The Mean Time Between System Failures)
$\bar{T}_s^{(F_s^1, F_s^2)}$, the mean time between failures F_s^1, F_s^2 is $\bar{T}_s^{(F_1,F_2)} = \bar{T}_s^F + \bar{T}_s^U$. $A_s = \frac{\bar{T}_s^F}{\bar{T}_s^{(F_s^1, F_s^2)}}$.

6.29 Definition (System Performability)
System performability, P_s, is the probability that a system's performance is above a certain level P_s^μ at a given instant of time t. Performance, unlike reliability, is a measure of the likelihood that some subset of functions is performed correctly with respect to a minimal baseline P_s^μ.

$D_s = f_2(R_s, A_s, P_s, M_s, S_s); R_s, A_s, P_s, M_s, S_s = f_1(F_s)$ so $D_s = f_3(F_s)$. System dependability probability is a function of system fault probabilities by composition of $f_2(f_1())$. To a first order approximation, $R_s = -k_1\Gamma_s, A_s = -k_2\Gamma_s, M_s = -k_3\Gamma_s, S_s = -k_4\Gamma_s, P_s = -k_5\Gamma_s$ where k_i are proportionality constants. System dependability is in general inversely proportional to system scale, to a first order approximation $F_s^p = k_6\Gamma_s$ and $F_s = k_7\Gamma_s$.

A basic design strategy to minimise k_1, k_2, k_3, k_4, k_5 is to incorporate state replication and persistence. Multiple subsystem replication at all system levels - network, node, micro-kernel, capsule and servers (see Appendix A) increases system dependability. Subsystem replication possibilities in this five dimensional space may have complicated topology. For instance, it may be appropriate to design replicated nodes, with hyper-cube network interconnection over replicated hardware communications, e.g. ISDN, to guarantee a single route in the case of one or several node failures and failure of a single ISDN channel of the critical open route. It may be appropriate to spread server replicas over nodes with Delaunay tessellation, i.e. shortest path between neighbours. The optimal global replica topology over network transport protocols, nodes, micro-kernels, and superstructural application capsules and clusters is application loading dependent. Hence optimisation of hard real time hierarchical fail

safe topologies is a super-problem of the previously defined coupled optimisation of event routing and application server loading problem. The optimal hard real time hierarchical fail safe replica topology is not necessarily the event routing optimal solution graph and the application loading optimal graph for minimum response time determined by (6.6). Micro-kernel persistence and replication increase system availability and reliability which becomes critical as a distributed object system system scales, i.e. Γ_s increases.

6.30 Definition (Application and Micro-kernel Persistence for High Availability and Reliability) If objects are persistent, if an object o_n fails, and an invocation fails, this will generate a system exception at kernel k_i to be returned to calling object o_m. Then the persistent object is reloaded ready for a second query if o_m retries. If a kernel k_i fails, then this is a serious fault. A persistent kernel protocol should also be defined. This may involves some level of logging of kernel backup states.

It is necessary to use FOCH to organise the replication into various granularities and use persistence for certain granularity state logging in order to use persistence back up at higher system levels, i.e. for complete subsystem failure. This occurs during FOCH design step 8.(b).

Persistence and replication increase P_s for a given P_s^μ but persistence and replication reduce P_s^μ because of persistence access overhead, transaction synchronisation of states, and increase in replica synchornisation traffic. Thus replication and synchronisation generally reduce P_s^μ for a given P_s but increase A_s and R_s. Replicas are not usually[20] localised on a single kernel to avoid kernel failure hence each system level of replication causes an increase Γ_s, for instance by doubling or tripling the network cabling, number of micro-kernels and objects.

One of the principle constraints on scalability is kernel processing performance. To simplify analysis, kernel response performance be described by:

1. $\bar{k}_i^e$, the kernel mean event processing response duration. This is the time between entry of an event from a transport protocol to the kernel and its exit via a transport protocol or via a superstructure processing stack.

2. $\bar{k}_i^p$, the kernel mean throughput. Throughput is equivalent to the inverse of inter-event arrival time over a measurement quantum.

3. $\bar{k}_i^a$, the kernel mean throughput absorption. This is the ratio of lost throughput and arrived throughput. This measure begins to be importance during scaling.

4. $\bar{k}_i^u$, the kernel utilisation rate. This is the ratio of active kernel event processing to kernel dormancy awaiting events.

5. $\bar{k}_i^q$, the mean length of the event queue on kernel k_i. As the kernel system scales, event queues[21] are introduced into the system wherever kernel utilisation rate is unity.

6. k_i^w, the kernel workload cycle time is the time interval between an event input and a previous event output of the kernel.

7. k_i^r, the kernel responsiveness time is the time interval between an event output and the previous event input to the kernel.

[20] The trivial replication case of software replication at the process level on a sequential processor is not often deployed.

[21] Kernel event queue processing design is an important issue. It may for instance be possible to temporarily fork a kernel to farm queue processing if there is a temporary bottleneck in processing. This strategy depends on the OS context switching scheduler but if kernels have processing priority over applications and the usual round robin context switching algorithm is in place, then a kernel fork could be useful. Alternatively, a kernel may increase its own priority to remove bottlenecks. Independent kernel determination of its own priority will cause all analytical scaling equations to be parameterised by kernel process scheduling priority.

8. k_i^c, a kernel cycle time is

$$k_i^c = k_i^w + k_i^r. \tag{6.14}$$

A basic performance scaling relation is that

$$k_i^c = \frac{1}{\bar{k}_i^p}. \tag{6.15}$$

Design requirements for throughput scaling can be converted to kernel cycle workload performance targets and responsiveness by (6.15) and (6.14). The goal of software engineering of the kernel system is to minimise for a given set of expected applications

$$\min(\bar{k}_i^e, \bar{k}_i^a, \bar{k}_i^q, k_i^c) + \max(\bar{k}_i^p, \bar{k}_i^u) \tag{6.16}$$

in order to maximise P_s for a given P_s^μ. All other system design factors being constant, D_s decreases as Γ_s increases. In particular, since $P_s = f_1(P_s^\mu), P_s^\mu = f_2(\Gamma_s)$, all other factors being constant, as Γ_s increases, for constant P_s, P_s^μ corresponding to P_s decreases. Conversely as P_s^μ increases, for fixed P_s, $\Gamma_s^{\max}$ increases. $\Gamma_s^{\max}$ defines the maximum scalability of a system for a given set of system dependability probabilities A_s, R_s, P_s, M_s, S_s. There is thus a complicated software engineering tradeoff between performance and dependability including performability. Dependability is increased by subsystem replication and persistence. Subsystem replication increases Γ_s but in this particular case, A_s, R_S, M_s are increased and P_s is decreased for a given P_s^μ.

Ω, a system design and engineering function, i.e. the function governing organisation of software engineering to generate distributed system implementation code takes as arguments a list of software developer resources $\vartheta_s = r_s^1, \ldots, r_s^p$, a software engineering methodology χ_c, and a system s requirements specification $\mho_s$, and generates system code $\omega_s = \Omega(\mho_s, \chi_c, \vartheta_s)$.

$$(R_s, A_s, S_s, P_s, C_s, \bar{T}_s^U, \bar{T}_s^{(F_s^1, F_s^2)}, F_s^a, F_s^b, F_s^c, \bar{T}_s^F) = f(\Omega(\mho_s, \chi_c, \vartheta_s)) \tag{6.17}$$

The goal of distributed system engineering is to generate the optimal dependable scalable system ω_s^* which satisfies

$$\max_{\chi_c, \vartheta_s}(R_s, A_s, S_s, P_s, C_s, \bar{T}_s^U, \bar{T}_s^{(F_s^1, F_s^2)}, \frac{1}{\bar{T}_s^F}, \frac{1}{\bar{T}_s^{(F_s^1, F_s^2)}}, \frac{1}{\bar{T}_s^U}) \tag{6.18}$$

where there is some suitable balance of dependability variable maximisation priority. The goal of system engineering is to $\max(D_s)$ where $D_s = f(\Gamma_s)$. A given ω_s and D_s imply a maximum Γ_s. While one can extistentially define (6.17) and (6.18), it is not in general possible to determine the functional[22] form of these equations. It is a difficult task to organise software engineering to generate ω_s^*. See §6.5, pg. 131. Unless all design and implementation elements consistently support the unity of architectural scalablity, the implementation and integration will not scale. An integrated system is only as scalable as the dependability of its least dependable element. All design decisions and implementation decisions must conform to maximisation of integrated system scalability. Software engineering management of implementation must sustain the unity of dependable scalable high performance system design as it expands from the architectural to the implementation level.

System scalability requires system tolerance of fault showers occurring simultaneously at all levels. Engineering multi-level scalable fault tolerance is in general wasteful of resource allocation and there

[22] The functional form of (6.17) is difficult to specify because of the many socio-technical variables that contribute to software engineering outcomes at the code level.

is a fault shower switch point where it is economically more viable to shut down the system gracefully and restart using SICO. An application developer should define graceful degradation of an application in the case of a hierarchical fault shower - simultaneous failure of routers, nodes and servers. In order to estimate and predict system performability probability P_s, it is necessary to model and predict system performance $P_s = f(\Gamma_s)$.

6.5.3 Hierarchical Modelling and Prediction of Micro-Kernel System Performance

It is is difficult to deepen the above outline of functional and non-functional requirements specification and progress from deepened requirements to design and engineering since in general a given set of requirements may not satisfiable by a design and a design may not be implementable on current hardware and software. Assuming that designs are feasible and implementable, a system implementation architect needs tools to assist design of models of in-existent components and performance measurement and calibration tools for locking implementations of optimised component models onto model performance reference targets during steps 8.(c) and 9 of the FOCH system design. Optimisation of unimplemented designs is defined in terms of an optimisation traversal of free parameters. Once components are implemented, free parameters are calibrated to peformance measurement data sets.

A subsystem performance model is a set of substem functions f_j that map a set of subsystem performance factors $\mathfrak{f}_i$ to a set of subsystem responses $\mathfrak{r}_j$. $\mathfrak{r}_j = f_j(\mathfrak{f}_1, \ldots, \mathfrak{f}_i)$. There are four main types of subsystem models f_j that map factors to responses:

1. algorithmic complexity models of deterministic code blocks,
2. analytical solution of product form probabalistic queuing models,
3. analytical approximate of solution of non-product probabalistic queuing models via:
   (a) decomposition and aggregation,
   (b) diffusion,
   (c) iterative methods,
4. numerical simulation:
   (a) Markov models,
   (b) Petri nets to represent delays due to queue synchronisation.
5. non-parametric approximation of implemented subsystem performance calibration data.

The first three methods are applicable to modelling of both unimplemented and implemented components. A distributed system's response probabilites are a function of the system processing input distribution and the probable respones of system components during processing of the input distribution. The probability response distribution of a system with a single processing queue as a function of the input load probabality can be integrated analytically if clean analytic approximations of input load probability, e.g. Poisson, are used. The output response probabalities of queing models containing recursive data transformation loops between component outputs and inputs, for instance models of operating system process schedulers, and/or queing system models which contain multiple queues which input into other queues are in general difficult to integrate explicitly. Hence approximations and numerical simulation is employed. The results of numerical simulations of coupled recursive multi-queue systems can be stored in a nonparametric regression approximator such as a neural network.

It is difficult to analyse and simulate the performance of distributed systems because of system complexity. Since typically a basic system design may have of the order 100 design performance factors and 50 design performance[23] response. It is difficult to calibrate a very large performance model and also difficult to undertake a complete approximation of a monolithic factor-response performance model $\mathbb{R}^{100} \rightarrow \mathbb{R}^{50}$. An adequate coverage of this space by sampling each factor into 10 quanta and for instance using 10 trial simulation per quantum if an analytical model has not been derived in the general case instead of calibration data (or possibly averaged calibration data) would require of the order of 100^{100} system simulations.

In order to simulate complex distributed multi-tier computing systems it is necessary to decompose system modelling into five hierarchical modelling granularities. The organisation of these models is represented by

$$S_f^r = C_s^f(SS_d^b(T_c^{SS_d^b}(C_a^{SS^b}(s_1, s_n, h_1, h_m)), I_d^b), \ldots, SS_d^z(T_c^{SS_d^z}(C_a^{SS^z}(s_1, s_n, h_1, h_m)), I_d^z)) \quad (6.19)$$

where S_f^r is a system scenario f response r, C_s^f is system scenario f configuration model, SS_d^b is subsystem b scenario d, $T_c^{SS_d^b}$ is transaction scenario c of subsystem b, $C_a^{SS_b}$ is configuration a of subsystem b, s_n is software component model n and h_m is hardware model m, and I_d^b is an input distribution to subsystem scenario d and similarly for other subsystems z in system configuration f.

Each of the five modelling levels may contain up to 30 factors and it is not usual to be interested in up to sixty response modalities $\mathfrak{r}_m$. System responses may also be hierarchically decomposed into various statistical measures of system response time and processing volume throughput such as average, maximum, minimum. A system model contains models at five levels:

1. h_m hardware models for instance of a computer. This contains a CPU, memory, and disk models. An other important hardware component is the network. The factors of hardware models generally measure

   (a) factors such as device access_time, device maximum throughput, device memory, and input load.

   (b) responses such as throughput and processing duration.

2. s_n software component models for instance, databases, graphical user interfaces and micro-kernels. Typical model elements are:

   (a) factors - multi_thread level, management probes on/off,

   (b) responses - $\bar{k}_i^e$, $\bar{k}_i^p$, $\bar{k}_i^a$, $\bar{k}_i^u$, $\bar{k}_i^q$, k_i^w, k_i^r, k_i^c.

3. $C_a^{SS_b}$, a subsystem configuration graph models specifies the loading of software components onto hardware component models. A configuration graph only has factors describing the interaction of software components and the loading of sofware onto hardware components.

4. $T_c^{SS_d^b}$, a subsystem transaction is defined by a traversal of the software configuration loading graph between software and hardware models. For a given subsystem input factor distribution, maximal, minimal, and average transaction duration may be defined for a given transaction input distribution.

5. SS_d^b, a subystem scenario defined on a subsystem transaction graph has:

[23] For instance at the kernel design level, there are at least 8 kernel performance responses $\bar{k}_i^e$, $\bar{k}_i^p$, $\bar{k}_i^a$, $\bar{k}_i^u$, $\bar{k}_i^q$, k_i^w, k_i^r, k_i^c defined on pg. 144.

(a) factors - maximum synchrounous input load, minimum input load, average input load distributions, error conditions. Input load probability distributions are usually modeled by:

i. uniform random distributions,

ii. Poisson distributions,

iii. special purpose Markov distributions.

(b) responses - maximal, minimal and average system output response duration and throughput.

System scenario responses SS_f^r are defined on a configuration C_s^f of subsystem scenarios SS_d^b. For system transaction scenarios, it may be convenient to use standardised system transaction benchmarks such as the Transaction Council C and D Benchmarks after decomposition of system transactions into subsystem transactions. See (Mullender, 1989). The performance analysis phase 8.(c) of the FOCH system design methodology involves the following sequence of analyses:

1. definition of system, subsystem, transaction, and component performance requirements:

   (a) duration constraints,

   (b) throughput constraints,

   (c) throughput absorption constraints,

   (d) utilisation rate constraints.

2. definition of design models, factors and responses:

   (a) hardware component models,

   (b) software component models,

   (c) a subsystem configuration software loading onto hardware graph,

   (d) the set of subsystem transactions,

   (e) scenarios of subsystem transaction sequences,

   (f) system scenario combinations of subsystem scenarios.

3. definition of performance evaluation elements - performance model and model results,

4. implementation of component, subsystem, and system models,

5. optimisation of system, subsystem, and component performance models,

6. software engineering of source code to meet performance model targets,

7. definition of software performance measurement probes in executables and of system and measurement elements to collect model calibration data:

   (a) performance measurements of the input distribution model,

   (b) measurements of system, subsystem and components processing duration, throughput, throughput losses and utilisation rate.

System measurements of distributed system performance are usually performed with a standard system management platform that supports performance measurement via standard operating system hooks[24].

Standard second order optimisation[25] of simulated systems and components can optimise the components if the system or subsystem and transaction models support gradient evaluations or alternatively use simulated annealing techniques can be used if the models have discontinuities.

The goal of FOCH system specification performance modelling is to identify, analyse and eliminate system and subsystem scenario processing bottlenecks since performance bottlenecks are the main cause[26] of system scaling failure. An analysis of a performance bottleneck may reveal that more hardware components are required, different software components are required or how to reengineer partly reusable software components, a subsystem reconfiguration is required or a subsystem transaction must be modified.

At a lower frequency, after implementation of several systems of a system family and after gathering large amounts of system model calibration data, system performance models may be used to optimise software components, such as micro-kernels. It is also possible to do this without implementation experience but a priori assumptions about critical model factors that impact system scalability may be incomplete.

Since software technologies change rapidly, the design process should respond to the development of novel components. The advantage of abstract design is that it is durable and only requires technology available at a given instant on the market, e.g. CORBA, to meet its reference performance targets. Also if an optimised design cannot be implemented with commercially available hardware and software components, then the results of the design process provide developers with specific component performance targets for each component as it is developed. A numerical component performance specification provides an exact implementation performance reference target for developers in addition to usual functionality specifications.

FOCH system design factors in simplification of software performance analysis and modelling into software design. Simplification of system verification as priority over performance optimisation decisions at the implementation level. It is counter-productive at the systems level if a combination of high performance components interacts to generate a low performance system that cannot be analysed because of the ad hoc implementation optimisation of components. If during the FOCH step 11., the system verification phase, the depolyed system does not meet performance targets, the fall back response is to return to the performance simulation model.

6.5.4 Critical Superstructural Operations Interfaces to Micro-Kernels

The CORBA concept of low level distributed object services and vertical domain frameworks constructed on the services can be reused with Java-broker[27] technology especially if the Java RMI broker infrastrure is based on CORBA. It is convenient to class all vertical domain application development frameworks into the class of application superstructural components. Superstructural components can

[24] For instance in UNIX: `vmstat` - virtual memory statistics; `iostat` - input output device statistics; `sar` - system activity reporter; `ps` - process scheduler statistics.

[25] Since these are high dimensional design problems, a numerical supercomputer is required. Above a certain dimensionality, approximation of the Hessian for optimisation is not tractable and for instance conjugate gradient minimisation or matrix inversion is tractable (3k hours) up to about $64k \times 64k$ matrix inversion problems on special purpose parallel matrix inversion hardware such Connection Machine 2 running conjugate gradient matrix inversion.

[26] It is also important for large system integration contracts to accurately estimate software and hardware component costs of the solution in order to make a profit.

[27] This generic term does not refer to the Visigenics Java-broker product although the goals of this product are similar.

themselves be layered horizontal hierarchically with vertical domain specific refinements and application specific interfaces at the highest superstructural level of refinement. It is also useful to examine some of the specification lacunae of CORBA in order to identify key service interactions a distributed Java-broker kernel must support to take the load of scalabel mission critical enterprise computing superstructures.

Dependable scalable micro-kernel support for secure real-time non-blocking nested two phase commit query transactions on replicated persistent objects is a basic foundation for implementation of manufaturing superstructures. See Appendix B.

During the product design phase of manufacturing of large products, for instance aeroplanes, up to several hundred designers may be designing subsystems using different types of CAD data, for instance 2D and 3D wireframe CAD, exact solid CAD, surface CAD, shape deformation CAD, geodesic CAD, finite element CAD, and assembly modeling CAD. CAD data is generally large, for instance geodesic data sets for a wing section may contain several hundred thousand atomic data types. Design version control requires a centralised master design database to enforce design consistency during the parallel design activities of the designers. The master design database may have a second tier of specialist CAD server stations serving clusters of design client applications. Thus design activities may involve up to three tier data synchronisation. A transaction synchronised replica proxy object cache that can be updated by a dynamically created interface is required to support scalable three-tier STEP system design synchronisation. If a CAD client station queries a CAD server of one of the specialist CAD servers queries the master design database, for instance using a meta-query interface based on EXPRESS, the return of arbitrary results of a CAD query to the query application requires a dynamic interface. See pg. 231. Since CAD datasets are so large, local cache of the query results are required in order to avoid very large edit and edit update traffic between a proxy and its original. Similarly, if a CAD server edits on a remote master object at high frequency and the edit updates must be frequently passed from the server to the remote editing subsystem, it is necessary to cache the data set being edited on the server. In a single tier cache synchronisation system, a simple pessimistic transaction synchronisation protocols may support replica cache synchronisation - a deny all transactions except one pessimistic locking protocol combined with a cache delta propagation after cache original modification. If two caches compete for a lock on a data set, lock assignment can be determined by time stamp or by transaction priority or by network distance of transaction origin.

The introduction of CAD server and master design database replication for high system availability requires a two phase commit transaction protocol to support the deny all but one transaction but generate ACID of this transaction on replica databases before update-delta synchronisation of replica proxies. See pg. 232.

With two tier data synchronisation, for definition of scalable system level transactions, it may be necessary to introduce multi-level transaction protocols and transaction protocol selector components to switch transaction mechanisms according to various optimisation criteria. For very large systems, to avoid the bottlneck of transaction manager synchronisation, it is possible to assign transaction control to active transaction aware data elements that select which transactions they permit from queued requests from multi-level transaction queues. In this case, additional logic must be introduced into the system to maintain global consistency between local transaction aware data elements selection of transactions from queues.

Support for these superstructural operations should be factored into the *ab initio* design, i.e. the interface definitions and the service base class implementations, of the Java-broker. Service interaction

scenarios are the use-case[28] context of caching, replication, synchronisation, transaction, persistence, query and security interfaces to the Java-broker kernel. The kernel must support the service interface combination semantics. The non-combinatoric simplest case interface to the services must support the most complex combinatoric possibility without developer recourse to workaround hacking at the RMI definition level. If analysis is deep enough, it may be possible to generate a system integration API that cleanly and concisely expresses complex object information processing dynamics at the object level that is also isolated from underlying operation systems (except JavaOS) and non-Java implementation languages. A FOCH designed Java-kernel that supports efficient compactly expressed complex service combinations allied to a refined FOCH superstructural systems engineering methodology provides a foundation for engineering of scalable distributed manufacturing superstructures.

6.5.5 A Software Pattern for Porting the Micro-Kernel

While it is possible to implement a Java-broker that executes on a Java virtual machine, broker scaling performance requirements may require direct broker binding to operting systems, transport protocols and other language bindings such Active X. Efficient porting is a hard low level problem which can have a great impact on high level software architecture. The implemenation architecture of Java-broker must be designed to contain porting hot spots in FOCH step 8.(a).

When a Java-broker is ported to support many operating systems, target binding languages, and transport protocols, the source code can fragment into many branches off the main source tree. The branches are difficult to merge and it is difficult to add new functionality into all branches. It is difficult to simultaneously include new features across Java-broker binding to all target operating systems, binding languages, and transport protocols. Multiple port source branches off the primary functional source trunk at different functional locations generate many subtle interacting bugs which are difficult to systematically eradicate. This section proposes a software architectural design pattern for implementation of Java-broker that supports porting to a variety operating systems and binding languages and transport protocols.

The main implementation work of porting a Java-broker occurs during checking of header files and in the pre-processing phase where header files are compared to system files. Porting of a Java-broker based on ad-hoc nesting of conditional pragmas quickly lead to unmanageable unstable inextensible unreusable unreliable code. Ad hoc conditional port pragma confusion of core function Java-broker code also distracts from readability of basic functionality.

The main software engineering trick to solve the Java-broker porting problem is to generate a core code base η that compiles across target operating systems, compilers, and transport protocols. For micro-kernel development, it is necessary to wrap non-portable code ζ in indirection wrappers. It is required to maintain centralised control over the core portable kernel code. There are three main classes of non-portable code:

1. operating system calls:
   (a) sockets,
   (b) i/o, - file system, terminal input,
   (c) signals,
   (d) threads,

[28] For instance, one use case of the security service is that it is involved in a multi-level secure transaction synchronisation of replica cache objects returned by a query on an untrusted meta-data repository. If the scenario complexity is not designed into the base classes, it will not magically appear 500,000 lines later.

(e) shared memory calls,

(f) environment query,

(g) interface language tokens.

2. transport protocols such as:

(a) Transport Control Protocol/Internet Protocol,

(b) Transport Control Protocol/Universal Datagram Protocol,

(c) Common Management Information Protocol,

(d) Simple Network Management Protocol,

(e) Integrated Standard Digital Network,

(f) Hyper Text Transfer Protocol,

(g) Frame relay,

(h) Profibus.

3. compilers.

Developer experience with compilers on different operating systems is required in order to analyse portable language constructs. For instance, in order to implement an efficient event kernel transport core that is easy to port across operating systems, transport protocols, and compilers, a deep study is required of how to define a common multi-thread class across operating systems with mono-thread a simple default. The interaction of multi-threading and inheritence is tricky. Multi-threading requires that mutual exclusion locks and thread synchronisation semaphores. Usually the developer is required to check these semaphores and locks in implementation source. The most critical point is when the thread schedular assigns a thread to a recently unlocked data element. The first action of the freshly awaken thread is to lock the data element. Thus there are two levels at which a multi-thread compiler can operate

1. all data elements may potentially be multi-threaded in which case as many locks as the number of data elements are required. Lock search can be a serious bottleneck in this case for fine granularity objects.

2. only flagged data elements are the object of thread competition for locks.

In both cases making static inheritence multi-threading transparent is straightforward. In both cases, the compiler generates the lock table. Hence without special lock conditions, the compiler can insert pre and post conditions thread operation blocks. However runtime polymorphism and operator overloading makes transparent locking tricky. In this case it is necessary to dynamically link, generic locking code, to the polymorphic threads into a suitable insertion point before thread execution.

In a server process, there is also a hierarchical arrangement of threads. The master thread processes commmunication requests and spawns primary threads for processing of input requests. Secondary threads may by spawned by primary threads to process individual elements of complex requests. Thus there is a three level hiearchy of threads. In the case that primary or secondary threads contain identical processing sequences on a given data element in memory, the thread mutual exclusion problem occurs. In the case that a primary slave thread spawns n secondary sub-threads, the primary thread must synchronise on the results of all of the secondary threads before passing the results to the master communication thread. The return communication between a primary and master thread is

itself a synchronisation problem defined on a buffer. If two primary threads close to simultaneously synchronise on their secondary threads, then each will compete for a mutual exclusion lock on the results communication buffer that is read FIFO by the primary communication thread after it acquires a read lock on the result buffer. It is required to define a base class that makes all of this processing transparent to the application server programmer. If a developer specifies the server object and methods, the thread mutual exclusion and sychronisation base class should robustly convert all derived classes to multi-threaded operation which is tested by a query to the operating system at runtime to determine the maximum if any number of threads which the operating system can supply with single process the degenerate processing case.

A thread synchronisation and mutual exclusion base class must be a single inheritence route. Multiple inheritence with other base classes containing asynchronous event processing loops, such as graphical user interface codes, will cause multi-thread operation anomalies if these have not been engineered to be thread safe. A concurrent object compiler should for non real-time applications remove the visibility of threading from the developer except where he requires explicit control of thread priority, or error handling, for instance of two phase commit nested transactions on server objects. It is not possible to shield a developer from explicit thread control in a real-time concurrent object compiler but the developer should be provided with intuitive constructs such as `process(lock(object,lock_conditions))`. Some design hints for generation of a portable multi-thread base class are given by for instance the Mach virtual multi-thread implementation.

It is easier for a programmer to localise porting dependencies of an OS s_j or transport protocol t_l if these are located in seperate individual files. If a developer has to insert a new porting dependency, then an object oriented build pre-processor translates that dependency into the the correct hierarchical conditional wrapper before compilation by the target compiler onto the target operating system for support of the target transport protocol.

Let core portable kernel code be organised in a kernel directory with sub-directories k_i for each of the segmented function module groups. For porting development it is necessary to have a directory for each operating system s_j with directories for each compiler c_k and further directories for each transport protocol t_l. Unfortunately, the porting space $\{s_j, c_k, t_l\}$ is not orthogonal, i.e. there may be subtle interactions between (s_j, c_k, t_l). An object oriented build is required to cope with these porting interactions. An object oriented build also maps well to an object component software engineering and testing methodology. There are thus two levels of object oriented build:

1. build of object components up to the system build level,

2. build across target operating systems, transport protocols and target binding languages.

Both of these levels can be supported by build tools such as `imake`. The principle software engineering porting goal is to maintain η independent of s_j, c_k, t_l and to give an application developer a single view of each of s_j, c_k, t_l for the purposes of porting wrapper localisation. Assuming for the moment orthogonality of (s_j, c_k, t_l), once portability analysis of portable core code η for instance based on POSIX calls vs port dependency peripheral code ζ has been completed, it is straightforward to wrap s_j dependent calls by one level of indirection. It is difficult and time consuming to wrap a compiler for a given s_j and a t_l for a given c_k but less difficult than navigation of a conditional port pragma tree over the whole source code base. The porting developer should thus see three files for a given port operation. If there are no s_j, t_l, c_k interactions, then port dependency isolation and generation of robust wrappers requries exhaustive behavioural testing. If there are s_j, t_l, c_k interactions, then it is required to define a fourth file $i_{j,l,k}$ containing interaction wrappers. The interation wrappers $i_{j,l,k}$

generate a second level of wrapper indirection, i.e. they wrap the port collision constructs in s_j, t_l, c_k. The build sequence is thus:

1. If $i_{j,l,k}$ exists, $i_{j,l,k}$ pre-process s_j, t_l, c_k to s'_j, t'_l, c'_k else $s'_j = s_j$, $t'_l = t_l$, $c'_k = c_k$.
2. If s'_j, t'_l, c'_k exist these pre-process the port dependency source ζ to $\zeta^{j,l,k}$.
3. then the complete code base $\mu = \eta + \zeta^{j,l,k}$ or component z, $\mu_z = \eta_z + \zeta_z^{j,l,k}$ is compiled by c_k to execute on s_j.

The port directory structure is thus:

1. OS s_j:
   (a) sub-dir - UNIX POSIX
      i. vendor 1 extensions,
      ii. $\vdots$
      iii. vendor n extensions,
   (b) sub-dir NT ....
2. Compilers c_k:
   (a) C++ standard,
      i. vendor 1 extensions,
      ii. $\vdots$
      iii. vendor n extensions,
   (b) Ada ....
3. Transport Protocols t_l:
   (a) ISO/OSI tcp/ip:
      i. vendor 1 extensions,
      ii. $\vdots$
      iii. vendor n extensions,
   (b) Profibus ...,
   (c) tcp/udp,
   (d) CMIP,
   (e) SNMP,
   (f) Profibus,
   (g) ISDN,
   (h) Frame relay,
   (i) http.
4. Port interactions,
   (a) interaction j, l, k,
   (b) interaction $j + 1, l, k$,

(c) ....

Once a heterogeneous application development design, implementation, compile and test cycle is up and running, code development moves forward across all $\{s_j, c_k, t_l\}$ at once albeit at a slower pace than moving the code forward on a single (s_j, c_k, t_l) followed by a delay wave of porting activity to all (s_j, c_k, t_l) by generating new source branches full of conditional pragmas and trying to merge them occasionally in order to understand what is core code. Without an object oriented build, a port source tree has two levels of branches from it, the first level is functional extension branches from the basic stable functional trunk. The second level of branches is the porting level. An object oriented build prunes the second level of branches and makes possible simultaneous releases across all operating systems, transport protocols and compilers of new Java-broker features.

Merging of functional code branches into the trunk should occur in strict sequence of size of change merging with rigorous testing of code functionality at finer granularity than a single code development as merge progresses. At the software quality assurance stage, each functional branch is tested before core remerge and after core remerge. The trunk has an inverted pyramid shape as a function of time in terms of code volume and number of functions/features added. Periodically, it is necessary to engage in a complete FOCH system analysis to compact the volume where possible.

In order to add a new s_j or c_k or t_l to a object build configuration system, a core code freeze occurs on the main source trunk. The new s_j or c_k or t_l is analysed and tested until it compiles to the core code freeze point. It is tested on the code freeze point. When it passes the tests on the code freeze point, it is tested on the latest state of the functional remerge sequence. When it passes this test, it is absored into the compile wave of (s_j, c_k, t_l). In order to maintain a pure portable source core, the code development cycle is thus:

1. code a new functional branch rigorously using basic portable syntactic constructs of compilers rather than advanced features.
2. use a single (s_j, c_k, t_l) compile to generate a working model followed by a
3. distributed[29]. compile with all (s_j, c_k, t_l) and gather all the error reports from each compile sequence.
4. fix bugs,
5. analyse dependencies and generate interaction wrapper $i_{j,k,l}$.
6. restructure η and ζ.
7. analysis of the distributed compile across all (s_j, c_k, t_l) combinations may result in update of (s_j, c_k, t_l) and all port interaction files,

It is also useful to isolate linguistic tokens involved in user interaction, e.g. GUI labels, in a single file to facilitate linguistic porting. For maximum international portability to target natural languages, it is useful to employ unicode lexical representations in user interaction functions.

6.6 Scalable Open Distributed Agile Manufacturing Systems

In conclusion, Java-ODP based distributed manufacturing systems become more attractive as Java components and services, e.g. system management, events, transactions, persistence, and security are

[29] Distributed compile is supported by Clearcase

perfected. Java currently requires a scalable broker system for its RMI interface. If a scalable Java-ODP remote invocation broker could be implemented satisfying the above preliminary functional and non-functional requirement specification, then large integrated manufacturing systems could perhaps be constructed with:

1. Java application development tools and Java beans,
2. event transport core supporting Java virtual machines RMI,
3. an ISO/OSI and Profibus transport protocols,
4. CORBA wrap of legacy OS and programming languages.

An ODP based manufacturing application superstructure constructed on the distribution infrastructure would contain components that support at least the following elementary manufacturing functions:

1. management:
   (a) strategic and tactical business objective management,
   (b) system management:
      i. communications,
      ii. computers,
      iii. control of infrastructure operations of communications and computers. control has:
         A. synchronous sub-systems,
         B. asynchronous sub-systems.
      iv. staff and line personnel.
   (c) data management:
      i. geometric model management based on EXPRESS/STEP and metadata models,
      ii. assembly metadata management,
      iii. materials resource planning.
   (d) process and operations management:
      i. manufacturing order processing management,
      ii. requirements specification management,
      iii. design management,
      iv. process model management,
      v. process error diagnosis and recovery management,
      vi. device drivers and device management,
      vii. distribution and logistic system management.
2. scheduling:
   (a) concurrent pipelined design, engineering and manufacturing,
   (b) inventory flows,
   (c) production scheduling, e.g. JIT Kahnban. See (Harrison, 1992).
   (d) production process visualisation and simulation.

3. decision support:
   (a) expert systems,
   (b) statistical Engines - e.g. SAS, SPSS, SPlus,
   (c) numerical engines - e.g. Matlab, Gauss,
   (d) production process visualisation and simulation.
4. software engineering:
   (a) components development master strategy and architecture revision control,
   (b) software visualisation,
   (c) distributed software system simulation and model calibration,
   (d) Software CAD and pre-implementation design optimisation,
   (e) distributed software engineering,
   (f) distributed build of distributed build targets,
   (g) source revision control and branch merging strategies,
   (h) integrated coding and document generation,
   (i) software verification and validation.
5. manufacturing scenario configuration:
   (a) application definition and configuration,
   (b) sensor and general device configuration,
   (c) software maintenance,
   (d) manufacturing sytem performance modelling, measurement and optimisation.
6. back office operations:
   (a) billing and cost analysis,
   (b) marketing analysis,
   (c) settlement,
   (d) video-conferencing.

Most of these manufacturing functions were beyond the implementation scope of the prototype environment. Analysis of the design of large manufacturing systems reveals that the larger the systems become, the more complicated it is to calibrate the total system to physical production reality measured by sensors. Calibration of either parametric or nonparametric models of manipulator kinematics embedded in functionalist or object or behaviourist components of architectures is a fundamental problem for grounding large flexible STEP and Profibus based sensor and model intensive distributed robotics and automation systems. It is necessary to calibrate these large flexible manufacturing control softwares to workspaces. The basic model calibration issues explored in previous Chapters are relevent to the development of higher levels and more rapid automation of the pipeline from custom manufacturing order processing, requirements specification, product design to rapidly reconfigurable flexible robotic parts production, assembly, product wrapping, and input into distribution systems.

The general conclusion after implementation and validation of the distributed object environment for prototyping of manipulator control algorithms in Chapter 6 is that the main bottleneck in development of higher levels of manipulator functionalities is extraction of low-level invariant features from signals. It is also necessary for system designers and developers to work at the mathematical level with tools to support more automatic compilation of mathematical models to the distributed infrastructure details. Robot manipulators are elements of distributed manufacturing systems. Advanced manufacturing requries management of hundreds of thousands of devices and computational objects. Development of high performance large scale distributed computation satisfying soft real-time constraints of high system availability and reliability requires tool and library software engineering infrastructures to enable developers to easily implement reliable synchronous and asynchronous sub-computations, multi-tier reliable communications, system management, transactions, replication, fault-tolerance, authentication and security. Construction of abstract control shells on top of distributed partially asynchronous devices is desirable for control algorithm prototyping but abstraction of hardware by specification of universal drivers for a complex device sub-class such as processing of sequences of video camera images reduces execution efficiencies and complicates integration of device drivers.

Signal data fusion, compression, feature extraction and robust classification (Horn, 1986) into discrete pre-defined categories or patterns for input into planning systems based on STRIPS (Fikes and Nilsson, 1971), i.e. DFA, connects symbols used at manufucturing mission symbolic planning and management level to signals. Symbols can be logically defined in two ways

- **intensionally** by rules for classifying symbols, for instance $\{x | 1 \leq x \leq 10, x \in \mathbb{Z}\}$.
- **extensionally** by enumeration of symbol elements, for instance $x = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

Symbolic processing on computers is implemented by operations on integers with extrinsic semantics attached by observers. It is unlikely that a stable general task and context invariant representation of the world, and by inference manufacturing superstructure systems, can be derived for encoding into formal representations to be operated on by symbolic logic systems. See (Penrose, 1989; Dreyfus, 1979; Putnam, 1981; Searle, 1981; Varela et al., 1991; Smith, 1991). (Putnam, 1987) proved the indeterminacy of model theory, the canonical formalisation of automatic symbolic planning and mangement. For any model-theoretic definition of the relationship of symbols to meaning, (Putnam, 1987) demonstrates that if semantics characterises meaning, then semantics does not characterise the way symbols relate to entities in the world. In model theory, if S is a sentence composed of predicates $P_i, i = 1, \ldots, n$, each with a value of 1 or 0 (i.e, true or false), according to model-theory, the meaning of S is a function that assigns a truth value $\{0, 1\}$ to that S in each possible situation. The meaning of each term in a sentence is its referent in each possible situation. Changing the meanings of the predicates P_i changes the meanings of S for a theory of meaning to be consistent. If this requirement is violated by a valid sentence S_* governed by a theory T of meaning, T is refuted. (Putnam, 1987) proves that the model-theoretic meaning, i.e. $\{0, 1\}$ value, of sentence S_* a cat is on a mat is constant while the referent of P_1 cat is changed from cats to cherries and the referent of P_2 mat is changed from mats to trees. If the reference of the parts changes while the truth of S_* in every interpretation is preserved, (Putnam, 1987) infers that the pairing of meaningless strings of symbols with meaningless structures cannot provide a theory of meaning. The (Putnam, 1987) proof of the indeterminacy of model theory reference implies that as analysis of assembly operations in more unstructured task-spaces becomes more complex, it is increasingly difficult to derive appropriate inter-locking symbols and models used by automatic symbolic planning, fault diagnosis and recovery reasoning systems that

match with invariant categories output from sensor processing. A priori prescriptions can not anticipate all contingencies that could arise during a given interaction of a robot with a workspace. This causes automatic symbolic planning and management algorithmic failures of representational rigidity and brittleness. This general epistemic constraint makes it difficult to specify and imlemement object software components for robotics and automation that are reusable across a wide variety of processing domains. The potential advantages of objected oriented software development:

1. increased software reusablity,
2. reduced software labour costs,
3. more rapid system development and deployment,

may fail to materialise because of analysis and design failures caused by:

1. Restriction of requirements analysis scoping. This causes non-reusability of components,
2. Context specification failures. This can cause inter-object operation failures,
3. Incorrect object granularities. This reflects incorrect system analysis,
4. Opaque class designs. This can result from lack of experience or isolated code development,
5. Lack of commercial components,
6. Greater inffeciency of codes compared to declarative codes because of the cost of virtual table dispatch of polymorphic functions,
7. Good object designers are rare.
8. greater difficult to learn than declarative programming. Good object programmers are expensive.

The initial FOCH design of an object system has a critical impact on the reusability of the final components. In object oriented software development it is more difficult to separate the design and implementation functions compared to the cascade software lifecycle of structured systems analysis and design methods. With object orientated systems development it can be more difficult to abstract design out from implementation and design is sometimes influenced in its early and final stages by the range of compiler language features, e.g. in C++. System architecuture designs are actualised and deepened by manyimplementation level design decisions. The architecture design expands into the implementation phase. The final architecture is the implementation. If design has been correctly structured, the formal system architecture is a shallow description of the final system architecture.

To construct large distributed robotics and automation systems, top-down systems analysis can be combined with object system analysis. FOCH systems without inheritance or polymorphism between subtype objects above a certain hierarchical composition granularity are easier to analyse and verify at the systems integration level. Large inheritance and polymorphic object systems can not in general be adequately analysed above a certain system complexity level. Hence their dependability cannot be verified. At the system level of analysis, it is appropriate to define a functional interface to object sub-systems to restrict the calling graphs between these large object sub-systems. This introduces the classic system functional rigidity that facilitates analysis and restricts system re-engineering. If a good functional split of object sub-systems is defined, system re-engineering to incorporate evolving technological substrates can be confined to the object subsystems while preserving a stable higher level

functional decomposition. Thus object subsystem components can be debugged, analysed, verified, maintained and re-engineered with minor effect on the system analysis at the functional level.

The correct application of a refined FOCH method may give good results, fast, at high expense. Robotics and automation projects can fail if FOCH analysis is flawed at any level and/or team engineers can/do not communicate clearly and adequately system assumptions and analysis interfaces. A simplified FOCH software engineering method, for instance without the performance simulation, can be applied to specification of a matrix organisation for software engineering. At the systems engineering methodological level, the technical limits of the FOCH methodology are circumscribed by both:

1. the limitations of functional and object system engineering methods.
2. the scaling problems of matrix organisation of software engineering.
3. individual engineer incapacity to introspect semantic primitives and imagine action contingencies, i.e. limitations of analysis.

In general, engineering results scale sub-linearly in the number of engineers that must be coordinated due to communication failure, engineering sub-system isolation, and development path sequence, structural and functional interdependence. It is difficult to maintain the architecture in the presence of temporal, logical, and structural inconsistency in decision making that can occur if organic conflicts (e.g. psychodynamic, personality, gender, age, functional, regional, project, line management vs staff, operations vs line management) lurking in a matrix organisation decision making and operations become 'cancerous', i.e. self-sustaining decision islands.

As software engineering scales, the more developers are assigned to problem solution, the more time they must spend communicating with each other both verbally and by writing and reading sub-system documentation in order to ensure successful system integration. The second technical circumscription of systems engineering possibilities is the introspective imaginative analytical limitations of each individual engineer during design. This leads to the incorporation of introspective categorical bias (c.f. (Kelly, 1955)) into robotic and automation task representations. An engineer assumes that environmental dynamics and sensor processing correspond to this introspective modelling bias. An engineer must assume that modelling bias does not jeopardise model calibration to workspace objects. These assumptions hold until system failure. Failure probability is a function of system complexity.

Many robotics and automation system vendor's basic strategy is to develop their application development toolkits with interfaces to STEP, MMS, CCE, Profibus, CORBA, DCE, DCOM, and Java standards. A CORBA to MMS mapping is defined in (Usländer, 1995b). By inference it is possible to map Java to Profibus. It is difficult to engineer a high performance Java-Broker with several protocol stack interfaces to distributed heterogeneous computing systems.

The integration of massively parallel servers into a dependable scalable distributed object system increases the class of strategic resource and materials assignment problems that can be solved, i.e. it expands the class of practical effective computable functions.

6.31 Definition (The Set of Computable Functions)

The set of computable functions is that set of functions that can be computed by a universal Turing[30] machine (Davis, 1958).

[30] (Turing, 1936) defined computation formally as a sequence of discrete states of an imaginary 4-tuple machine as it moves along an imaginary instruction tape. Machine states are determined by a set of rules and instructions encoded on the tape. A Turing machine computation (Turing, 1936) is a sequence of 4-tuple machine states as the machine moves along the tape. Any computer design can be simulated by a universal (Turing, 1936) computer.

The set of computable functions is a logical concept. The set of computable functions is independent of hardware devices. The set of computable functions does not depend on the existence of physical computers except for a fundamental computer, a mathematician. There are many interesting questions about this class such as whether an empirical sub-partition between non-polynomial time for deterministic function evaluation is equal to polynomial time for function evaluation, i.e. is P=NP?

6.32 Definition (The Set of Effective Computable Functions)
The set of functions for which an algorithm exists for function evaluation. It is at this level of computation that most programmers operate mentally, i.e. at the algorithms and data structures level in order to expand the class of Defintion 6.33.

The set of effective computable functions is also independent of the existence of physical computers although certain subsets of this set may be defined with respect to different models of computers.

6.33 Definition (The Set of Practical Effective Computable Functions)
This is the set of effective computable functions for which hardware exists for function evaluation within a problem life span, say one to ten years.

The set of practical effective computable functions is hardware and software dependent by definition. Distributed parallel[31] computations are useful for expansion of the set of effective and practical effective computable functions. See (Bertsekas and Tsitsiklis, 1989). Sets of parallel and distributed computation have no special properties with respect to the set of computable functions.

An important component of more advanced nuclear and space platform systems are high performance dextrous manipulators connected to complicated sensor processing control systems. Dextrous manipulators can simultaneously avoid trajectory singularities common with non-redundant manipulators, maximise dynamic variation over $\mathfrak{W}$, and maximise manipulability M (see §2.4) while operating in restricted or cluttered environments, optimise path planning kino-dynamic constraints such as minimisation of energy or acceleration jerk. To solve these multi-objective functions in real-time, efficient evaluation of all solutions of the inverse kinematics of dextrous manipulator are required to be computed by a cyclic inversion algorithm. The (Manocha and Canny, 1994) general solution algorithm based on evaluation of the eigendcomposition of transformed matrix polynomials described in §2.5.3 is reported to have a solution evaluation time for arbitrary dextrous manipulator of $10ms$ on RS\6000. This is fast enough for many real-time on-line Cartesian trajectory inversion problems and it can be executed in parallel on separate processors to compute simultaneous multi-objective solutions. After elementary differential topology definitions in the next Chapter, Chapter §8 proposes a dextrous manipulator solution method, which may be more efficient than (Manocha and Canny, 1994) algorithm, based on geometric analysis of the solution groups of dextrous manipulator homomorphisms

[31] The distinction between parallel and distributed computation lies in the granularity and synchronicity of component computations running on meshes of processing elements. If computations are fine grain synchronous they are typically classified as parallel. If computations are coarse grain asynchronous they are classified as distributed.

Chapter 7

Solution Groups of a Homomorphism between Lie Groups

Manipulator kinematic mappings are defined between continuous compact[1] connected[2] metric spaces that have group structure. Groups are an algebraic abstraction of transformation invariance, i.e. symmetry. See (Weyl, 1952; Weyl, 1946). Metric topology classifies equivalence classes of deformations of manifolds (surfaces) defined on continuous metric spaces (Gemignani, 1972). Lie groups are manifolds that have group structures. The forward kinematic function $\mathcal{F}$ is invariant under a set of transformation groups G_k defined on sub-spaces of image and pre-image variables. The invariance of a kinematic mapping under these transformations is due to both the group and the topological structure of T^n and $\mathsf{SE}(3)$. The solution group of a generalised system of equations is the largest local group of transformations acting on the pre-image and image variables with the property that solutions of the system are transformed to other solutions (Olver, 1993). This chapter defines metric topological properties of Lie groups and homomorphisms between Lie groups in order to derive solution groups that act on a local linear representation of a solution element of a dextrous manipulator homomorphism in Chapter 8.

7.1 Manifolds

7.1 Definition (A Metric Space)

Let $\mathbb{X}$ be any set. $(\mathbb{X}, D)$ is a metric space if a metric function $D : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ is defined on $\mathbb{X}$ such that:

1. $D(x, y) \geq 0, \forall x, y \in \mathbb{X}$,
2. $D(x, y) = D(y, x), \forall x, y \in \mathbb{X}$,
3. $D(x, y) = 0 \iff x = y$,
4. $D(x, y) + D(y, z) \geq D(x, z), \ \forall x, y, z \in \mathbb{X}$.

[1] $\mathsf{SO}(3)$, T^n are compact. A compact space is 'bounded and finite'.

[2] A space $\mathbb{X}$ is connected if $\nexists \mathbb{Y} \subset \mathbb{X}, \nexists \mathbb{Z} \subset \mathbb{X}$; $\mathbb{Y}, \mathbb{Z}$ open sets, such that $\mathbb{X} = \mathbb{Y} \cup \mathbb{Z}$ and $\mathbb{Y} \cap \mathbb{Z} = \emptyset$ and $\mathbb{X} \neq \emptyset$ and $\mathbb{Y} \neq \emptyset$.

7.2 Definition (A Homeomorphism)

f, a function from a metric space $(\mathbb{X}_1, D_1)$ to a metric space $(\mathbb{X}_2, D_2)$, is a homeomorphism if f is injective, surjective, and continuous, and if f^{-1} is continuous.

7.3 Definition (A Diffeomorphism)

If $f : \mathbb{X} \subset \mathbb{R}^n \rightarrow \mathbb{Y} \subset \mathbb{R}^m$, a map between two subsets of two Euclidean spaces, is bijective and smooth, and $f^{-1} : \mathbb{Y} \rightarrow \mathbb{X}$ is also smooth, then $\mathbb{X}$ is diffeomorphic to $\mathbb{Y}$ and f is a diffeomorphism.

7.4 Definition (A Manifold)

If $\mathbb{M}^n$ is a subspace of a space $\mathbb{R}^g, g > m$, $\mathbb{M}^n$ is an m-dimensional **manifold** if it can be approximated in a sufficiently small neighbourhood of any $x \in \mathbb{M}^n$, by a mapping to $\mathbb{R}^m$, a **chart**.

7.5 Definition (A Chart)

A chart (U, ϕ) is a diffeomorphism ϕ of an open subset U of $\mathbb{M}^n$ to an open subset of $\mathbb{R}^m$. For any $q \in U$,

$$\phi(u(q)) = (x_1(u(q)), \ldots, x_m(u(q))). \tag{7.1}$$

The m smooth functions $x_1, \ldots, x_n$ are called the coordinate functions or more simply the coordinates of q in the chart. ϕ^{-1} is a parameterisation of U.

7.6 Definition (An Atlas)

A set of intersecting charts $\{U_\alpha, \phi_\alpha\}$ whose domains U_α cover $\mathbb{M}$ is called an atlas. A manifold $\mathbb{M}^n$ can be represented by a union or atlas of charts.

7.7 Definition (An Immersion of a Submanifold)

A submanifold of $\mathbb{M}$ is a subset $\mathbb{N} \subset \mathbb{M}$, together with a smooth, injective map $\phi : \tilde{\mathbb{N}} \rightarrow \mathbb{N} \subset \mathbb{M}$ satisfying a maximal rank[3] condition everywhere, where the parameter space $\tilde{\mathbb{N}}$ is some other manifold and $\mathbb{N} = \phi(\tilde{\mathbb{N}})$ is the image of ϕ. In particular the dimension of $\mathbb{N}$ is the same at that of $\tilde{\mathbb{N}}$, and does not exceed the dimension of $\mathbb{M}$. This method of defining a submanifold is known as an immersion.

7.8 Definition (An Implicit Submanifold)

A submanifold $\mathbb{N}$ can also be defined implicitly. If $\mathbb{M}$ is a smooth m-dimensional manifold and $F : \mathbb{M} \rightarrow \mathbb{R}^n, n \leq m$ is a smooth map and F is of maximal rank on the subset $\mathbb{N} = \{\mathbf{x}_i : F(\mathbf{x}_i) = \mathbf{0}\}$, then $\mathbb{N}$ is a regular, $(m - n)$ dimensional submanifold of $\mathbb{M}$.

7.1 Theorem (The Global Embedding Space of a Manifold (Whitney))

Every m-dimensional manifold $\mathbb{M}^n$ can be embedded in at least $\mathbb{R}^{2m}$. This means that every element of $\mathbb{M}^n$ has an open neighbourhood that is diffeomorphic to an open subset of a single chart of $\mathbb{R}^{2m}$. The bijective application that maps any element of $\mathbb{M}^n$ to its representation in $\mathbb{R}^{2m}$ is called an embedding of $\mathbb{M}^n$ in $\mathbb{R}^{2m}$. $\mathbb{R}^{2m}$ is called the ambient Euclidean space of $\mathbb{M}^n$. Once an atlas is defined, $\mathbb{M}^n$ can be represented as a collection of copies of $\mathbb{R}^m$ and real-analysis defined in vector spaces can be applied to mappings between manifolds by defining tangent spaces to the manifold without referring to its embedding in $\mathbb{R}^{2m}$. This collection of tangent spaces is called a fibre bundle.

7.2 Fibre Bundles

7.9 Definition (A Fibre Bundle)

A fibre bundle $\mathfrak{FB}$ is a tuple $(\mathfrak{B}, p, \mathfrak{F}, \mathfrak{W})$ where $\mathfrak{B}$ is called the total space, $\mathfrak{W}$ is called the base space, $\mathfrak{F}$ is called the typical or canonical fibre, and p is a projection $p : \mathfrak{B} \rightarrow \mathfrak{W}$, with a set of open coordinate

[3] A mapping F between manifolds $\mathbb{M} \rightarrow \mathbb{N}$ of dimension m and n respectively is maximal rank at $\mathbf{x}_i \in \mathbb{M}$ if the rank of the Jacobian $\mathbf{J}$ at $\mathbf{x}_i$ of F is equal to m or n, whichever is smaller.

neighbourhoods W covering $\mathfrak{W}$ such that there exists a family of local homeomorphisms $\phi_W : p^{-1}(W) \rightarrow W \times \mathfrak{F}$, with $W \in \mathfrak{W}$, where $\mathbf{x}_i = p(\phi_W^{-1}(\mathbf{x}_i, \theta_i))$ with $\mathbf{x}_i \in W$ and $\theta_i \in \mathfrak{F}$.

ϕ_W are coordinate functions for neighbourhoods W. ϕ_W functions parameterise W. The set $\theta_i = p^{-1}(\mathbf{x}_i)$ is called the fibre above $\mathbf{x}_i$, and is homeomorphic to a canonical fibre $\mathfrak{F}$. Thus θ_i denotes the collection of points on the fibre above $\mathbf{x}_i$. It is common to refer to a fibre bundle by its total space $\mathfrak{B}$.

7.10 Definition (A Trivial Fibre Bundle)
A fibre bundle $\mathfrak{FB} = (\mathfrak{B}, p, \mathfrak{F}, \mathfrak{W})$ is called trivial if $\mathfrak{B} = p^{-1}(\mathfrak{W})$ and $\mathfrak{B}$ is isomorphic to the product space $\mathfrak{W} \times \mathfrak{F} = \phi(\mathfrak{B})$, where $\mathfrak{B}$ maps to $\mathfrak{W}$ in a canonical way, $\phi(\mathfrak{B}) = \mathfrak{W} \times \mathfrak{F} \rightarrow \mathfrak{W}$. In this case, there is a canonical projection, $\pi : \mathfrak{W} \times \mathfrak{F} \rightarrow \mathfrak{W} : \mathbf{x}_i = \pi(\mathbf{x}_i, \theta_i)$, such that $p = \pi(\phi())$. $\mathfrak{B}$ is called locally trivial if every $\mathbf{x}_i \in \mathfrak{W}$ has a neighbourhood W such that the fibre bundle $(\mathfrak{B}|_w, p, \mathfrak{F}, W)$ is trivial.

All fibre bundles are locally trivial. If $p^{-1}(\mathfrak{W})$ is globally a product space, $\mathfrak{B}$ is a trivial fibre bundle.

7.11 Definition (The Contractability of a Trivial Fibre Bundle)
A fibre bundle $(\mathfrak{B}, p, \mathfrak{F}, \mathfrak{W})$ is trivial if $\mathfrak{W}$ is contractable. See (Burdick, 1988).

Open simply connected subsets of $\mathbb{R}^n$ are contractable. If a fibre bundle has $\mathfrak{W}_w, w = 1, \ldots, r$ as a simply connected open region of the workspace, it is a trivial fibre bundle. Spatial and dextrous manipulator kinematic mappings are sets of trivial fibre bundles defined on maximal singularity free simply connected open regions of the workspace between $\mathbb{CPS}$. A base space $\mathfrak{W}$ can be decomposed into a union of r maximally $\mathbb{CPS}$-free sub-spaces $\mathfrak{W}_w$ plus q $\mathbb{CPS}$. $\mathfrak{W} = (\cup_{i=1}^{r} \mathfrak{W}_w) \cup (\cup_{a=1}^{q} \mathbb{CPS}_a)$.

7.12 Definition (A Cross Section of a Fibre)
A cross section of a fibre bundle, $\{\theta_i | \theta_i = \Upsilon^{-1}(\mathbf{x}_i), \mathbf{x}_i = p(\theta), \mathbf{x}_i \in \mathfrak{W}\} \subset \mathfrak{B}$, is defined by a continuous function $\Upsilon^{-1} : \mathbf{x}_i \in \mathfrak{W} \rightarrow \theta_i \in \mathfrak{B}$ such that $\mathbf{x}_i = p(\Upsilon^{-1}(\mathbf{x}_i))$. If a fibre bundle is trivial, then all cross-sections have the same topological structure. Any cross-section is an inverse function Υ over the base space $\mathfrak{W}$. Conversely an inverse function Υ^{-1} precisely defines a cross-section of the fibre bundle $p^{-1}(\mathfrak{W})$. All fibre bundles have a cross section.

7.13 Definition (A Direct Inverse Function Parameterisation of a Trivial Fibre Bundle)
If a fibre bundle is trivial, then cross sections through that bundle define a direct inverse function Υ^{-1} over the base space $\mathfrak{W}$. A sub-region $\mathfrak{W}_w \subset \mathfrak{W}$ is invertible if it contains no $\mathbb{CPS}$ and $p^{-1}(\mathfrak{W}_i)$ is a fibre bundle. A direct inverse function is a continuous function Υ_w^{-1} on a maximally singularity free subset $\mathfrak{W}_w$ of a workspace $\mathfrak{W}$, $\Upsilon_w^{-1} : \mathfrak{W}_w \rightarrow \{\theta_i\}$, such that $\mathbf{x}_i = p(\Upsilon_w^{-1}(\mathbf{x}_i)), \forall \mathbf{x}_i \in \mathfrak{W}_i$ (Wampler, 1988; Baker and Wampler, 1988). If an inverse function Υ_w^{-1} (Wampler, 1988) exists for $\mathfrak{W}_i$, it is invertible Υ_w. If a workspace region $\mathfrak{W}_w \subset \mathfrak{W}$ contains no $\mathbb{CPS}$, then $\forall \mathbf{x}_i \in \mathfrak{W}_w, \Upsilon_w^{-1}(\mathbf{x}_i)$ are all diffeomorphic fibres.

Υ_w^{-1} functions may be constructed in a number of ways. Assume that the fibre is embedded in a space $\boldsymbol{\theta}$ of dimension n and the dimensionality of base space is $m = \dim(\mathfrak{W})$ with $n > m$. A direct parameterisation $\zeta = \theta_{m+1}, \ldots, \theta_n$ of the forward function Υ_w from $\theta_1, \ldots, \theta_m$ to $\mathfrak{W}_w$ is invertible

$$\theta_1, \ldots, \theta_m = \Upsilon^{-1}(\mathbf{x}_i, \theta_{m+1}, \ldots, \theta_n), \tag{7.2}$$

possibly in closed-form. Appendix E derives a closed-form direct inverse function parameterisation of the inverse kinematic solution fibre bundle of planar 3-**R**. This work develops a nonparametric approximation of a direct inverse function parameterisation of the inverse kinematic solution fibre bundle of planar 3-**R** and 7 *dof* dextrous manipulators in Chapter 8. Υ_w^{-1} may also parameterise disjoint fibres.

7.14 Definition (A Disjoint Fibre)

A fibre is disjoint if $\{\theta_i | \theta_i = \Upsilon_w^{-1}(\mathbf{x}_i, \zeta),\ \forall \mathbf{x}_i \in \mathfrak{W}_w\}$ is not connected. If a fibre is disjoint, the number of fibre branches is defined to be the number of maximally connected subsets of $\Upsilon^{-1}(\mathfrak{W}_w, \zeta)$.

A disjoint fibre has disconnected pieces. A disjoint fibre is hence not contractable. Fibre bundles considered in this work are locally trivial fibre bundles but globally their topology is complicated by the fact that fibres $\mathfrak{F}_1, \mathfrak{F}_2$ over different subsets of base space $\mathfrak{B}$ can belong to different topological classes. No global direct inverse function exists over the whole of $\mathfrak{W}$. A disjoint fibre $\mathfrak{F}$ on $\mathfrak{W}_w$ is homeomorphic to a contractable fibre $\mathfrak{F}$ on $\mathfrak{W}_w$ but the fibres are not diffeomorphic. Fibre contractability and disjointness are homotopy properties.

7.15 Definition (The Homotopy of Fibres)

If $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are two fibres, $\mathfrak{F}_1 \subset \mathbb{M}, \mathfrak{F}_2 \subset \mathbb{M}$, are homotopic if they are diffeomorphic to a manifold $\mathbb{K}$. Let the diffeomorphisms be denoted $f_1 : \mathfrak{F}_1 \rightarrow \mathbb{K}$, $f_2 : \mathfrak{F}_2 \rightarrow \mathbb{K}$. Let t be a coordinate of a chart on $\mathbb{K}$. $\mathfrak{F}_1$ is homotopic to $\mathfrak{F}_2$ if

$$\exists F : \mathbb{K} \times [0,1] \rightarrow \mathbb{M}, \tag{7.3}$$

$$F(t,0) = f_1^{-1}(t) \text{ and } F(t,1) = f_2^{-1}(t). \tag{7.4}$$

where F is continuous. f_1^{-1} and f_2^{-1} are pullback functions from $t \in \mathbb{M}$ to $\mathfrak{F}_1$ and $\mathfrak{F}_2$. F is said to be a homotopy between $\mathfrak{F}_1$ and $\mathfrak{F}_2$.

Homotopy generates equivalence classes of submanifolds in a topological space. Two fibres $\mathfrak{F}_1, \mathfrak{F}_2$ are homotopic if they can be mapped to each other by a diffeomorpism. The inverse kinematic solution fibres of Planar 3-$\mathbf{R}$ and the 7 *dof* dextrous manipulators analysed in Chapter 8 are fibres that are homeomorphic to S^1. The S^1 fibres are embedded in T^3, the 3-torus, in the case of Planar 3-$\mathbf{R}$ and T^7 in the case of 7 *dof* manipulators. Closed S^1 fibres embedded in T^3 or T^7 may belong to an infinite set of homotopy classes because from zero to an infinite number of non-contractable loops around a torus are possible for a S fibre.

Some manifolds are invariant under groups of transformations. For instance a sphere is invariant under an axial rotation. If $\mathbf{x}_i$ is a point on the surface of a sphere and the sphere is rotated by η around an arbitrarily selected polar axis, then $\mathbf{x}_i$ is mapped to $\mathbf{x}_j$, with $\mathbf{x}_j$ also on the surface of the sphere. A polar axial transformation of the sphere is a continuous invariant transformation of the sphere that maps the sphere onto itself. A continuous transformation group defined on a manifold is a Lie group (Olver, 1993).

7.3 Lie Groups

7.16 Definition (A Group)

A group $(\mathsf{G}, \oslash)$ is a set G of elements together with a group operation $\oslash$ that satisfies the following axioms

7.1 Axiom (Closure)

For any two elements g and h of G, the product $k = g \oslash h$ is also an element of G.

7.2 Axiom (Associativity)

If $g, h, k \in \mathsf{G}$, then $g \oslash (h \oslash k) = (g \oslash h) \oslash k$.

7.3 Axiom (Identity Element)

$\forall g \in \mathsf{G}, \exists i \in \mathsf{G}$ such that $i \oslash g = g = g \oslash i$.

7.4 Axiom (Inverse Element)
$\forall g \in \mathsf{G}, \exists g^{-1}$ such that $g \otimes g^{-1} = i = g^{-1} \otimes g$.

Examples of well known groups are $\mathsf{G} = (\mathbb{R}\backslash\{0\}, *)$, $\mathsf{G} = (\mathbb{R}, +)$ and $\mathsf{G} = (\mathbb{Z}, +)$.

7.17 Definition (An Abelian Group)
$(\mathsf{G}, \otimes)$ is an abelian group if an additional axiom is satisfied:

7.5 Axiom (Commutativity)
$\forall g, h \in \mathsf{G}, g \otimes h = h \otimes g$.

If G is a discrete set of elements, $(\mathsf{G}, \otimes)$ is a discrete group. If G is a continuous set of elements, $(\mathsf{G}, \otimes)$ is a continuous group.

7.18 Definition (A Homomorphism)
A homomorphism is a mapping $\varsigma : \mathsf{G} \rightarrow \mathsf{F}$ between two groups $(\mathsf{G}, \otimes), (\mathsf{F}, \oplus)$ which respects the group operations

$$\varsigma(g_a \otimes g_b) = \varsigma(g_a) \oplus \varsigma(g_b), \ g_a, g_b \in \mathsf{G}. \tag{7.5}$$

If $\exists \varsigma^{-1}$, a smooth inverse, there exists ρ an isomorphism between G and F.

7.19 Definition (A Product Group)
If $(\mathsf{G}, \otimes)$ and $(\mathsf{F}, \oplus)$ are two different groups, then a product group $(\mathsf{GF}, \boxtimes)$ with $\boxtimes$ defined for $g \in \mathsf{G}, f \in \mathsf{F}$ is defined by:

$$\forall g_a, g_b \in \mathsf{G}, f_c, f_d \in \mathsf{F}, (g_a, f_c)(g_b, f_d) = (g_a \boxtimes f_c, g_b \boxtimes f_d). \tag{7.6}$$

(7.6) is abbreviated to $\mathsf{G} \boxtimes \mathsf{F}$.

7.20 Definition (A Cyclic Group)
If g_o is a subset of G, the smallest subgroup of G containing g_o is written $\mathsf{G}(g_o)$. A group is cyclic if there exists an element $g_o \in \mathsf{G}$ such that $\mathsf{G} = \mathsf{G}(g_o)$. A cyclical group of order n is denoted C_n.

7.21 Definition (A Subgroup)
A subgroup H of G is a subset of elements of G such that $(\mathsf{H}, \otimes)$ also satisfies axioms 7.1-7.4 and possibly axiom 7.5 with H substituted for G.

7.22 Definition (The Left and Right Cosets of a Subgroup)
If H is a subgroup of G and $g \in \mathsf{G}$, the set $\mathsf{H}g$ (the set of all elements $hg, \forall h \in \mathsf{H}$), is called a right coset of the subgroup H in the group G. Similarly for left cosets $g\mathsf{H}$.

7.23 Definition (Group Divisors)
A subgroup H of a group G is called a normal divisor of the group G if $\forall g \in \mathsf{G}, g\mathsf{H} = \mathsf{H}g$. That is, $\forall g \in \mathsf{G}, h \in \mathsf{H}$, there exists an element $h_2 \in \mathsf{H}$ for which $gh_1 = h_2 g$. If G is abelian, every subgroup of G is a normal divisor of G. The group G itself and the identity subgroup i are trivial normal divisors of G. If they exist, all other normal divisors are non-trivial.

7.24 Definition (A Factor Group)
If H is a normal divisor of G, G/H is the factor group of G by the normal divisor H. Multiplication in the factor space G/H is defined by the coset $\mathsf{H}g_1g_2$ that is the product $(\mathsf{H}g_1)(\mathsf{H}g_2)$ of cosets $\mathsf{H}g_1, \mathsf{H}g_2$.

Definitions 7.16 to 7.24 combined with certain formal restrictions (see (Naimark and Stern, 1982; Olver, 1993)) define the factorisation of a product group $\mathsf{G} \times \mathsf{H}$ by a normal divisor H and a factor group $\mathsf{G/H}$. For factorisation of groups related to the kinematics of dextrous manipulators, it is useful to define

7.25 Definition (A Local Group)

A local group of transformations is a group without Axiom 7.1, group closure.

Local group omission of the group closure axiom makes group factorisation more flexible but extra criteria must be introduced for consistent definition of group factorisation without closure. See (Olver, 1993). Two groups that differ topologically by one element cannot be factored by a common group factor, then if one group is defined as a local group, then two groups may perhaps be factored by a common factor group and the group closure property has only one exception in the local group. This exception is topologically important since it changes the topological structure of one group which is why the groups could not be factored by a common factor in the first instance. Local groups without identity elements are also important in this work for factorisation of groups related to the kinematics of dextrous manipulators.

7.26 Definition (A Truncated Group)

If Definition 7.25 of a local group G as a group without Axiom 7.1 for group closure implies that G also does not satisfy Axiom 7.3, i.e. there is no identity element, then G is a truncated[4] group.

The truncated group removal of the closure axiom 7.1 and the identity element axiom 7.3 from the group axioms leaves the associativity axiom 7.2, the inverse axiom 7.4, and perhaps the commutivity axiom 7.5. The effect of the removal of the closure and identity element axioms is that general group theorems are no longer valid in the form of $\forall$ assertions for the truncated group. Group theorems may be valid for for most assertions of the form $\forall x_i \backslash a, b, c$, i.e, for all group elements except a,b,c. A truncated group can be more easily factorised than a group.

A manifold $\mathbb{M}^m$, for instance the sphere, may have discrete or continuous group structure. A Lie group L is a smooth manifold and a continuous non-abelian group such that the group operation is a differential mapping from $\mathsf{L} \times \mathsf{L}$ onto L (Olver, 1993).

7.27 Definition (An m-parameter Lie Group)

An m-parameter Lie group is a group L which also carries the structure of an m-dimensional smooth manifold in such a way that both the group operation

$$\xi : \mathsf{L} \times \mathsf{L} \rightarrow \mathsf{L},\ \xi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \oslash \mathbf{x}_j,\ \mathbf{x}_i, \mathbf{x}_j \in \mathsf{L} \tag{7.7}$$

and the inversion

$$\iota : \mathsf{L} \rightarrow \mathsf{L},\ \iota(\mathbf{x}_i) = \mathbf{x}_i^{-1},\ \mathbf{x}_i \in \mathsf{L} \tag{7.8}$$

are smooth maps between manifolds.

Let G be a local group of transformations acting on a Lie manifold L. A subset $\mathsf{A} \subset \mathsf{L}$ is called G-invariant if is is unchanged by the group transformations, i.e. $\mathbf{x}_j, \mathbf{x}_i \in \mathsf{A}$ whenever $\mathbf{x}_j \in \mathsf{G}$ and $\mathbf{x}_i \in \mathsf{A}$.

[4] This is a neologism since a standard term could not be found in the literature.

7.28 Definition (An Orbit of a Group)

An orbit $\mathbb{O}$ of a local transformation group G acting on a Lie manifold L is a minimal nonempty group-invariant $A \subset L$. $\mathbb{O} \subset L$ is an orbit if:

1. If $\mathbf{x}_j \in \mathbb{O}, \mathbf{x}_i \in G$ and $\mathbf{x}_i \otimes \mathbf{x}_j$ is defined, then $\mathbf{x}_i \otimes \mathbf{x}_j \in \mathbb{O}$.
2. If $\tilde{\mathbb{O}} \subset \mathbb{O}$, and $\tilde{\mathbb{O}}$ satisfies the above condition, then either $\tilde{\mathbb{O}} = \mathbb{O}$ or $\tilde{\mathbb{O}}$ is empty.

A Lie manifold can also be factored into a quotient manifold by a group.

7.29 Definition (A Quotient Manifold)

If G is a local group of transformations acting on L, there is an induced equivalence relation among the points $\mathbf{x}_h$ of L, with $\mathbf{x}_h$ being equivalent to $\mathbf{x}_l$ if they lie in the same orbit of G. L/G, a quotient manifold, is defined by the set of equivalence classes, or equivalently, the set of orbits of G. The projection $\kappa : L \rightarrow L/G$ associates to each $\mathbf{x}_h$ in L its equivalence class $\kappa(\mathbf{x}_h) \in L/G$ which can be identified with the orbit of G passing through $\mathbf{x}_h$. In particular, $\kappa(\mathbf{x}_h) = \kappa(\mathbf{x}_i \otimes \mathbf{x}_h)$ for any $\mathbf{x}_i \in G$ such that $\mathbf{x}_i \otimes \mathbf{x}_h$ is defined. Conversely, given a point $\mathbf{x}_j \in L/G, \kappa^{-1}(\mathbf{x}_j)$ will be the orbit determined by $\mathbf{x}_j$, realised as a subset of L. The quotient space L/G has a natural topology obtained by requiring that the projection $\kappa[U]$ of an open subset $U \subset L$ is open in L/G. L/G can be endowed with the structure of a smooth manifold if certain regularity conditions are specified on G. If L is an m-dimensional manifold and G has s-dimensional orbits, then the quotient manifold L/G will be of dimension $m - s$.

A quotient manifold is the continuous Lie group version of a factor space of a factor group G of Definition 7.24. This work factorises Lie groups represented canonically with respect to representations of the factor groups.

7.4 Representations of Lie Groups of Kinematic Mappings

7.30 Definition (A Representation of a Group)

D is a representation of the group G if there exists a homomorphism (see Definition 7.18) to a linear mapping in a vector space, the representation space R^n. After fixing a basis in R^n, $\forall g \in G, \exists D(g)$ such that $D \in R^n$ such that multiplication of the group elements $g, h \in G$ corresponds to $D(gh) = D(g)D(h)$. The representation is faithful if the homomorphism $g \rightarrow D(g)$ is one-to-one.

A group can have many representations. See (Naimark and Stern, 1982). For instance, a geometrical axial symmetry in $\mathbb{R}^2$ of the plane is isomorphic to C_2, the two cycle group. After choosing a basis and defining reflection matrices, an axial symmetry in $\mathbb{R}^2$ is a representation of the cyclic group C_2 under the action of reflection through the axis. Axial inverse kinematic solution symmetries of the plane and addition modulo 2 are representations of the same group. This section examines representations of Lie groups involved in manipulator kinematic mappings. From (2.7), $\mathcal{F}$ is a mapping from an n torus T^n to a m-dimensional $\mathbb{C}^m$ Cartesian space, usually the Special Euclidean group.

7.31 Definition (The Torus Group)

The n torus T^n has Lie group structure. $T^n = S^1 \times S^1 \times \ldots \times S^1$, an n-fold product of the circle S^1. S^1 has the Lie group structure of SO(2), the Special Orthogonal Group in the plane. (Olver, 1993). The product of Lie groups is also a Lie group, hence T^n is a Lie group. Each torus T^n is a connected, compact, abelian, n-parameter Lie group and, in fact, is the only such Lie group up to isomorphism (Olver, 1993).

7.32 Definition (The Special Euclidean Group)

The Special Euclidean Group[5] is the product group of $\mathsf{SE}(3) = \mathbb{R}^3 \times \mathsf{SO}(3)$. $\mathsf{SE}(3)$ is a connected Lie group. $\mathsf{SE}(3)$ has Lie structure since both $\mathbb{R}^3$ and $\mathsf{SO}(3)$ have Lie structure (Olver, 1993). $\mathsf{SE}(3)$ is the group underlying transformations of position and orientation of rigid bodies in three dimensional space. Each image of Cartesian space corresponds to a unique transformation of the Cartesian frame. The composition of two transformations ${}^a_b\mathbf{T}^c_d\mathbf{T}$ is the internal binary operation. For two elements of a representation of $\mathsf{SE}(3)$, ${}^a_b\mathbf{T}^c_d\mathbf{T} \neq {}^c_d\mathbf{T}^a_b\mathbf{T}$.

The Special Orthogonal Group was already mentioned in §2.1.

7.33 Definition (The Special Orthogonal Group)

$\mathsf{SO}(n)$, the Special Orthogonal group, is an n dimensional manifold with Lie group structure (Olver, 1993).

$$\mathsf{GL}(n) = \{\mathbf{X} : |\mathbf{X}| \neq 0, x \in \mathbf{X} \text{ and } x \in \mathbb{R}\} \tag{7.9}$$

$$\mathsf{O}(n) = \{\mathbf{X} \in \mathsf{GL}(n) : \mathbf{X}^T\mathbf{X} = I\} \tag{7.10}$$

$$\mathsf{SO}(n) = \{\mathbf{X} \in \mathsf{O}(n) : |\mathbf{X}| = +1\} \tag{7.11}$$

where GL is the General Linear group and $\mathsf{O}(n)$ is the Orthogonal group.

$\mathsf{SO}(n)$ can be represented in various ways. The direct way from the above Definition and from §2.1 is for $\mathsf{SO}(3)$ to be represented by the subset of $N \times N$ matrices in $\mathbb{R}^{n\times n}$ with orthonormal columns (and rows) and determinant $+1$. Orthogonal matrices are special if their determinant is $+1$. Hence the name Special Orthogonal Group. From this definition, the Special Orthogonal Group is a subgroup of the General Linear Group of n dimensional invertible matrices with real entries. $\mathsf{SO}(3)$ matrices can be mapped to vectors in $\mathbb{R}^6$ by selecting the first two columns (the third column can be obtained as the cross-product of the first two). $\mathsf{SO}(3)$ can therefore also be represented by vectors in $\mathbb{R}^6$ by a single chart. This also provides an instance of Theorem 7.1. A three dimensional Lie manifold $\mathsf{SO}(3)$ can be globally coordinatised by a single chart to an ambient embedding six dimensional Euclidean space which is itself a special submanifold of a 9-dimensional Euclidean space.

$\mathsf{SO}(n)$ is the group underlying n dimensional rotations. There is no diffeomorphism from an open set of $\mathbb{R}^n$ to the whole of $\mathsf{SO}(n)$ since $\mathsf{SO}(n)$ is compact and $\mathbb{R}^n$ is not. Thus every differentiable map from an open subset of $\mathbb{R}^n$ onto $\mathsf{SO}(n)$ has singular points where the differential of a map is not invertible. Four parameterisations of rotations in 3 space were enumerated in §2.1 - fixed angles, Euler angles, equivalent angle-axis and Quaternions. All of these representations of $\mathsf{SO}(3)$ have singularities. In the neighbourhood of a singular point, small changes in position on the $\mathsf{SO}(3)$ manifold result in large changes of representation parameters. At a representation singularity, the maximal rank condition of a chart does not hold.

Each parameterisation has drawbacks depending on the nature of its representation singularities. For instance, Euler angles can result in singularities for large rotations and they are not suitable for representing small orientation errors. Furthermore, it is difficult to compute parameters corresponding to the product of two rotations from Euler parameters representation of two rotations. Quaternions are generally the most flexible representation of rotation.

[5] Euler 'discovered' $\mathsf{SE}(3)$.

7.34 Definition (The Quaternion Group)

Quaternions $\mu = (a \in \mathbb{R}, \mathbf{b} \in \mathbb{R}^3)$, a mixture of scalars and vectors, with a non abelian group sum operator $\oplus$ and a non-abelian product operator $\otimes$ form an algebra

$$\mu_1 = a_1 + \mathbf{b}_1, \quad \mu_2 = a_2 + \mathbf{b}_2, \tag{7.12}$$

$$(a_1, \mathbf{b}_1) \oplus (a_2, \mathbf{b}_2) = (a_1 + a_2, \mathbf{b}_1 + \mathbf{b}_2), \tag{7.13}$$

$$\mu_1 \otimes \mu_2 = [a_1 a_2 - \mathbf{b}_1^T \mathbf{b}_2] + [\mathbf{b}_1 \times \mathbf{b}_2 + a_1 \mathbf{b}_2 + a_2 \mathbf{b}_1], \tag{7.14}$$

$$\forall \mu, \quad \exists I = 1 \in \mathbb{R} \quad \text{such that} \quad \mu \otimes I = \mu, \tag{7.15}$$

$$\forall \mu \neq 0, \quad \exists \mu^{-1} \quad \text{such that} \mu \otimes \mu^{-1} = I. \tag{7.16}$$

The set of unit ($a_1 = 1$) quaternions with unit norm $||\mu||$ with $\otimes$ as its law of composition forms the quaternion group Q (Samson et al., 1991).

The non abelian group of unitary quaternions can also be considered as the three dimensional sphere S^3 endowed with a group structure. For every rotation $\lambda \in \mathsf{SO}(3)$, there exists a unitary quaternion $\mu \in \mathsf{S}^3$. Thus quaternions may be used to define a mapping

$$f : \mathsf{S}^3 \rightarrow \mathsf{SO}(3) : \lambda = f(\mu). \tag{7.17}$$

Every rotation λ can be represented by a quaternion μ and the product of two rotations is represented by the product of corresponding quaternions. The map f is a continuous group homeomorphism and is onto. Its kernel is $\{\pm 1\}$ which means that there are two quaternions $(\mu, -\mu)$ for every rotation λ

$$\lambda = f(\mu) = f(-\mu). \tag{7.18}$$

The absence of a single chart from quaternions onto $\mathsf{SO}(3)$ is caused by rotations of π. S^3 the sphere is locally diffeomorphic to $\mathsf{SO}(3)$ but the application $\mathsf{S}^3 \rightarrow \mathsf{SO}(3)$ is not a diffeomorphism, and thus S^3 is not an embedding of $\mathsf{SO}(3)$ in $\mathbb{R}^4$. Two S^3_+ hemispherical mappings to $\mathsf{SO}(3) - \pi$ and charts from $\mathbb{R}^4$ to the S^3_+ hemispheres is an embedding. See pg. 38 of (Samson et al., 1991).

Despite the singularities of Quaternions and Euler angle representations of $\mathsf{SO}(3)$, these are most often used in robotics to represent $\mathsf{SO}(3)$ instead of 3×3 matrices since while matrices are convenient for computation, it is generally impossible for a technician to deduce the nature of a rotation by examining a matrix. (Samson et al., 1991; Fu et al., 1987; Craig, 1989) show how to convert between the different representations of orientation, including orientation parts of homogeneous transform matrices of §2.1.

7.5 Solution Groups of a Lie Homomorphism Defined by Manipulator Kinematics

$\mathcal{F}$ the forward kinematic function is a mapping from an n torus T^n to a m-dimensional $\mathbb{C}^m$ Cartesian space, usually[6] $\mathsf{SE}(3)$

$$\mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{SE}(3) : \mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i). \tag{7.19}$$

$(\boldsymbol{\theta}_i, \mathbf{x}_i)$ denote[7] pre-image and image group elements represented in some coordinate system. $\mathcal{F}$ is a mapping between Lie groups $\mathsf{T}^n \rightarrow \mathsf{SE}(3)$, i.e. $\mathcal{F}$ is a homomorphism.

[6] The following arguments about factorisation of $\mathsf{SE}(3)$ also apply to $\mathsf{SE}(2)$.

[7] $(\boldsymbol{\theta}_i, \mathbf{x}_i)$ can denote a group element or coordinate representations of a group element depending on the context.

7.35 Definition (A Lie Homomorphism)
A Lie homomorphism $\mathcal{F}$ is a homomorphism between two Lie groups. See Definition 7.18.

The kinematic mapping Lie homomorphism $\mathcal{F}$ is not an isomorphism since T^n and SE(3) are different groups. This implies that $\mathcal{F}$ mapping singularities exist, i.e. the pre-image of $\mathcal{F}^{-1}$ has $\mathbb{CPS}$.

7.36 Definition (An Implicit Definition of a Lie Homomorphism)
A homomorphism between Lie groups $\mathcal{F} : \mathsf{T}^n \rightarrow \mathsf{SE}(3)$ is defined implicitly by

$$F : \mathsf{T}^n \times \mathsf{SE}(3) \rightarrow \mathbb{R}^l : F_v(\mathbf{y}_i) = \mathbf{0},\ v = 1, \ldots, l \tag{7.20}$$

where F_v are smooth real-valued functions. The solution set of F is a Lie homomorphism manifold L.

This defines an alternative way of regarding a homomorphism $\mathcal{F}$ as a manifold L. A function graph is a manifold and each each point of the manifold may be covered by a chart where some coordinates of the chart represent $\boldsymbol{\theta}_i$ in pre-image of $\mathcal{F}$ and the remaining coordinates of the chart represent $\mathbf{x}_i$ in the image of $\mathcal{F}$. Sets of such points are in this sense solutions of the manifold defined implicitly by (7.20).

7.37 Definition (A Solution of an Implicit Definition of a Lie Homomorphism)
A solution $\mathbf{y}_i = (\mathbf{x}_i, \boldsymbol{\theta}_i), \mathbf{y}_i \in \mathsf{L}$, a Lie homomorphism defined implicitly by (7.20), is $(\boldsymbol{\theta}_i \in \mathsf{T}^n, \mathbf{x}_i \in \mathsf{SE}(3))$ such that $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$.

By Definition 7.36, $\mathcal{F}$ implicitly defines L as the solutions of a set of v homogeneous equations

$$F : \mathsf{T}^n \times \mathsf{SE}(3) \rightarrow \mathbb{R} : \mathbf{x}_i - \mathcal{F}(\boldsymbol{\theta}_i) = \mathbf{0}, \tag{7.21}$$

$$\equiv F_v(\mathbf{y}_i) = \mathbf{0} \text{ for } v = 1, \ldots, 6. \tag{7.22}$$

where $\mathbf{y}_i = (\mathbf{x}_i, \boldsymbol{\theta}_i)$. $F_1(\mathbf{y}_i), \ldots, F_6(\mathbf{y}_i)$ are smooth real valued coordinate invariant functions defined on the manifold L implicitly defined by $\mathcal{F}$. $F_v, v = 1, \ldots, 6$ are equations in $n + 6$ unknowns $y_1, \ldots, y_{n+6}$. $\mathsf{T}^n \times \mathsf{SE}^3$ is an embedding space of L, i.e. $\mathsf{L} \subset \mathsf{T}^n \times \mathsf{SE}(3)$. $\mathcal{F}$ from pre-image $\boldsymbol{\theta}_i$ to image $\mathbf{x}_i$ implicitly defines $\mathsf{L} = \{\boldsymbol{\theta}_i, \mathcal{F}(\boldsymbol{\theta}_i) | \forall \boldsymbol{\theta}_i \in \mathsf{T}^n\}$. A solution of F is $\mathbf{y}_i = (\boldsymbol{\theta}_i, \mathbf{x}_i) \in \mathsf{L}$ such that $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\theta}_i)$.

$\mathcal{F}$ is invariant under a set of transformation groups G_k defined on sub-spaces of image and pre-image variables. The invariance of a kinematic mapping under these transformations is due to both the group and the topological structure of T^n and SE(3). The solution group of a generalised system of equations is the largest local group of transformations G_k acting on the pre-image and image variables with the property that solutions of the system are transformed to other solutions (Olver, 1993).

7.38 Definition (A Solution Group of an Implicit Lie Homomorphism)
If G is a local group of transformations acting on an implicitly defined Lie homomorphism L, a subset $\mathbb{O} \subset \mathsf{L}$, is called G-invariant, and G is called a solution group of L, if whenever $\mathbf{y}_i \in \mathbb{O}$, and $g \in \mathsf{G}$ is such that $\mathbf{g} \otimes \mathbf{y}_i$ is defined, then $g \otimes \mathbf{y}_i \in \mathbb{O}$.

Let G be a solution group of L, a Lie homomorphism manifold implicitly defined by Lie homomorphism $\mathcal{F}$. G is a local group of transformations G acting on L such that G transforms solution elements $\mathbf{y}_i \in \mathsf{L}$ to other solutions $\mathbf{y}_j \in \mathsf{L}$ such that $\forall g \in \mathsf{G}$, if $g \oslash \mathbf{y}_i$ is defined, then $\forall g, g \oslash \mathbf{y}_i$ generates a solution orbit.

7.39 Definition (A Solution Element of a Solution Group)
If $\mathbf{y}_i \in \mathbb{O} \subset \mathsf{L}$, where L is a Lie homomorphism, $\mathbb{O}$ is a G-invariant orbit, then $\mathbf{y}_i$ is a solution element of a solution group G. $\mathbf{y}_i$ a solution element may be a point or a set of connected points of an implicitly defined Lie homomorphism manifold L.

An orbit of a local transformation group G_k is a G-invariant subset $\mathbb{O}_i$ of L. $\mathbb{O}_i$ is defined by a set of solutions or subvariety determined by the common zeros of a collection of smooth functions $F = (F_1, \ldots, F_6)$. If a solution element $\mathbf{y}_i \in \mathsf{L}$ is set of connected points acted on by a solution group G of L such that G transforms $\mathbf{y}_i \in \mathsf{L}$ to other solutions $\mathbf{y}_j$ such that $\forall g \in \mathsf{G}$, if $g \otimes \mathbf{y}_i$ is defined, then $\forall g, g \otimes \mathbf{y}_i$ generates a solution orbit.

7.40 Definition (A Representation of Solution Group Factors and Elements)
Solution group factors and elements are mapped to a coordinate system representation by charts coordinate functions ϕ_α of an atlas $\{U_\alpha, \phi_\alpha\}$. For the solution element $\phi_\alpha : (\mathsf{C}, \mathsf{T}) \rightarrow (\mathfrak{C}, \mathfrak{J}) : (\mathbf{x}_i, \boldsymbol{\theta}_i) = \phi_\alpha(\mathbf{x}_i, \boldsymbol{\theta}_i)$ where $(\mathbf{x}_i, \boldsymbol{\theta}_i)$ denotes either group elements or a coordinate system representation of these elements depending on the context.

It is important to distinguish the four types of inverse kinematic solutions:

7.41 Definition (An Inverse Kinematic Solution for a Point in Cartesian Space)
Given $\mathcal{F}_b^{-1} : \mathsf{SE}(3) \rightarrow \mathsf{T}^n : \boldsymbol{\theta}_i = \mathcal{F}_b^{-1}(\mathbf{x}_i)$, $\boldsymbol{\theta}_i$ is one solution of a possible $b = 1, \ldots, 16$ discrete solutions when $n \leq m$ or $\boldsymbol{\theta}_i$ is one solution of a fibre of solutions when $n > m$.

7.42 Definition (The Set of Inverse Kinematic Solutions for a Point in Cartesian Space)
The complete set of kinematic solutions for $\mathbf{x}_i$ is $\{\boldsymbol{\theta}_i\}, i = 1, \ldots, 16$ solutions for $n \leq m$ or the fibre of solutions when $n > m$.

7.43 Definition (A Solution of an Implicit Definition of a Lie Homomorphism)
Given $F_v : \mathsf{T}^n \times \mathsf{SE}(3) \rightarrow \mathbb{R} : \mathbf{x}_i - \mathcal{F}(\boldsymbol{\theta}_i) \equiv F_v(\mathbf{y}_i) = 0$, for $v = 1, \ldots, m$ an implicit definition of a kinematic homomorphism manifold L, a solution of the set of equations F_v is $\mathbf{y}_i = (\mathbf{x}_i, \boldsymbol{\theta}_i) \in \mathsf{L}$.

7.44 Definition (A Solution Element of a Solution Group of an Implicit Homomorphism)
If G is a solution group factorisation of a kinematic homomorphism manifold L, a solution element[8] of G is a subset of L on which G acts.

In general s different G^s factorisations of a given L may be possible. For instance if $\mathsf{L} \equiv \mathsf{T}^{12}$, L may be factored into $\mathsf{G}^1 \equiv \mathsf{T}^3 \times \mathsf{T}^4$ or $\mathsf{G}^2 \equiv \mathsf{T}^6 \times \mathsf{T}^2$ solution groups. Solution groups for both factorisations of G are defined on a single $\mathbf{y}_i = \boldsymbol{\theta}_i \in \mathsf{T}^{12}$. $\mathsf{G}^3 = \mathsf{T}^6$ is also a single solution group for T^{12} if a solution element is defined as a T^6. In general, there are three possible types of factorisations $\mathsf{G}^s = \prod_{k=1}^{c} \mathsf{G}_k^s$ of L defined by $\mathcal{F}$:

1. G_k^s that act only on the pre-image space T^n,
2. G_k^s that act only on the image space SE(3),
3. G_k^s that act on T^n and SE(3).

A given factorisation G^s of L may act on a solution element that is the union of all inverse kinematic solutions for a given $\mathbf{x}_i$ plus $\mathbf{x}_i$ itself. For instance, a redundant-by-1-*dof* manipulator may have an inverse kinematic solution set for $\mathbf{x}_i$ homeomorphic to S^1 where S^1 is embedded in T^n. In the case of a dextrous 7 *dof* manipulator, it may also be possible to factorise L defined by the same manipulator into a solution group G^s that acts on a solution element homotopic to $(\mathbf{x}_i \subset \mathsf{SE}(3), \mathsf{S}^1)$ where S^1 is embedded in T^7.

There are two approaches to factorising the solution groups of a homomorphism between Lie groups - algebraic and geometric. Both algebraic and geometric factorisation of $\mathcal{F}$ of dextrous manipulators were attempted in the thesis.

[8] In order to avoid possible confusion over which solutions are in question in chapter 9, a verbose construction 'solution group solution element' is used for this type of solution.

7.6 Algebraic Factorisation of Solution Groups of a Homomorphism

(Olver, 1993) develops a Lie algebra of solution groups of differential and algebraic equations after some further definitions. $\mathbf{v}$, a vector field, defined in a tangent bundle to a manifold L associates a vector $\mathbf{v}_{\mathbf{y}_i}$ to each point $\mathbf{y}_i \in \mathsf{L}$. A parameterised curve $\phi : \mathbb{R} \rightarrow \mathsf{L}$ is an integral curve of the vector field $\mathbf{v}$ if its tangent vector coincides with the vector field $\mathbf{v}$ at each $\mathbf{y}_i \in \mathsf{L}$. A vector field may be represented by a field exponentiation operator $\phi(t) = \exp(t\mathbf{v})\mathbf{y}_i$ parameterised by t passing through $\mathbf{y}_i = \exp(0\mathbf{v})$ at $t = 0$. $\exp(t\mathbf{v})\mathbf{y}_i$ is the infinitesimal generator of the flow on L corresponding to $\mathbf{v}$, i.e. $\mathbf{v}|_{\exp(t\mathbf{v})\mathbf{y}_i} = \frac{d}{dt}\exp(t\mathbf{v})\mathbf{y}_i$. If G is a group acting on L, the Lie algebra of L is the space of all invariant vector fields of G defined by the Lie bracket.

7.45 Definition (A Lie Bracket)
Given two vector fields $\mathbf{v}, \mathbf{w} \in \mathfrak{V}$, a set of vector fields, on a manifold L, their Lie bracket[9] is the vector field $[\mathbf{v}, \mathbf{w}]$ which satisfies $[\mathbf{v}, \mathbf{w}]f = \mathbf{v}(\mathbf{w}(f)) - \mathbf{w}(\mathbf{v}(f))$ for any smooth function $f : \mathsf{L} \rightarrow \mathsf{L}$ where $\mathbf{v}(f)$ denotes the application of a vector field to a function f such that $\mathbf{v}(f(\mathbf{y}_i)) = \frac{d}{dt}f(\exp(t\mathbf{v})\mathbf{x}_i)|_{t=0}$. The Lie bracket, $[,] : \mathfrak{V} \times \mathfrak{V} \rightarrow \mathfrak{V}$ is:

7.6 Axiom (Bi-linear)
$[\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2, \mathbf{w}] = \alpha_1[\mathbf{v}_1, \mathbf{w}] + \alpha_2[\mathbf{v}_2, \mathbf{w}], \forall \mathbf{v}_1, \mathbf{v}_2, \mathbf{w} \in \mathfrak{V}$ and $\forall \alpha_1, \alpha_2 \in \mathbb{R}$.

7.7 Axiom (Skew Symmetric)
$[\mathbf{v}, \mathbf{w}] = -[\mathbf{w}, \mathbf{v}], \forall \mathbf{v}, \mathbf{w} \in \mathfrak{V}$.

7.8 Axiom (Jacobi-Identical)
$[\mathbf{v}, [\mathbf{w}, \mathbf{z}]] + [\mathbf{w}, [\mathbf{z}, \mathbf{v}]] + [\mathbf{z}, [\mathbf{v}, \mathbf{w}]] = 0, \forall \mathbf{v}, \mathbf{w}, \mathbf{w} \in \mathfrak{V}$.

See (Olver, 1993). The Lie bracket between $\mathbf{v}, \mathbf{w}$ is the infinitesimal generator of the commutator of the two associated flows. If G is a group acting on a manifold L, a vector field $\mathbf{v}$ is G-invariant if it is unchanged by the action of any group element. Functions may be defined on L that are invariant to G actions.

7.46 Definition (A Lie Group-Invariant Function)
If G is a local group of transformations acting on a Lie manifold L, a function $F : \mathsf{L} \rightarrow \mathsf{F}$, where F is another manifold, is called a G-invariant function if $\forall \mathbf{y}_i \in \mathsf{L}$ and $\forall g \in \mathsf{G}$ such that $g \oslash \mathbf{y}_i$ is defined,

$$F(g \oslash \mathbf{y}_i) = F(\mathbf{y}_i). \tag{7.23}$$

A real-valued G-invariant function $F : \mathsf{L} \rightarrow \mathbb{R}$ is called an invariant of G. $F : \mathsf{L} \rightarrow \mathbb{R}$ is G-invariant iff each component F_v of F is an invariant of G.

Given a vector field $\mathbf{v}$ that is an element of G, $\exp(t\mathbf{v}) : \mathsf{G} \rightarrow \mathsf{G}$ denotes the associated flow parameterised by t. Applying the flow to the identity element e defines the one-parameter subgroup $\exp(t\mathbf{v}) \equiv \exp(t\mathbf{v})e$, the infinitesimal generator of the subgroup. The infinitesimal generator of a vector field is thus a generator of an associated subgroup action. Flows generated by vector fields are equivalent to a local group action of the Lie subgroup of the parameterisation on the manifold. If the application of a vector field to a function leaves the function invariant under infinitesimal changes in a local Lie group defined by the field's infinitesimal generator, this provides a criterion for determination of subspaces of a manifold corresponding to a local Lie group solution orbit (Olver, 1993).

[9] An example of a Lie bracket is the linear space of $n \times n$ matrices with the Lie bracket defined by matrix multiplication as $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$, where $\mathbf{A}, \mathbf{B}$ are $n \times n$ matrices.

7.47 Definition (The Infinitesimal Criterion of G-Invariance)
If G is a connected group of transformations acting on a manifold L, $F : \mathsf{L} \rightarrow \mathbb{R}$ is a G-invariant function iff $\mathbf{v}(f) = 0, \forall \mathbf{y}_i \in \mathsf{L}$ and every infinitesimal generator $\mathbf{v}$ of G.

(Olver, 1993) develops comprehensive procedures for determination of groups of solutions of systems of nonlinear partial differential equations after further definition of the prolongation of a fibre bundle representation of an algebraic equation system into a jet bundle space where general combinatoric partial differential equation cross terms are defined. Examination of equivalence classes of solution groups of classes of partial differential equations is of indirect relevance to this work. Differential equations define families of algebraic equations, i.e. equivalence classes of solutions to differential equations. (Olver, 1993) proves that determination of solution groups of algebraic equations is a sub-problem of determination of solution groups of differential equations.

(Olver, 1993) determines solution groups G_k^s for some algebraic functions by a series of "fairly routine calculations" (Olver, 1993) pg. 94. For algebraic functions, the infinitesimal criterion for invariance follows directly from definitions of how functions change under the flow generated by a vector field defined by a field generator and an exponentiated definition of a field. (Olver, 1993) shows how derivation of complicated nonlinear conditions for the invariance of submanfolds or function families under classes of group transformations can be replaced by an equivalent linear condition of infinitesimal invariance under infinitesimal generators of the differential equation solution subgroup actions. For a given equivalence class of algebraic equations, its solution group can be prolonged into a solution group defined on the jet bundle of the differential equations defining the algebraic equations. Determination of groups of solutions of algebraic equations is defined on field representations of the homomorphism, i.e. the image field induced by the Jacobian of the homomorphism and the pre-image field of pre-image manifold intrinsic curvature. Similar procedures are required to develop an exponentiation representation of the infinitesimal generator of the combined pre-image and image fields of the homomorphism. Invariant symmetric transformations of the infinitesimal generators determine the solution subgroup actions of the homomorphism.

An attempt was made in the thesis to apply the above outlined procedures to algebraic determination of solution groups of $\mathcal{F}$ of 7 *dof* dextrous manipulators. The solution group determination procedures defined in (Olver, 1993) for generating solution groups of homomorphisms defining differential equation equivalence classes are at quite an abstract level of generality often assuming a canonical representation of differential or algebraic equation systems. It is necessary to select a representation of the homomorphism in order to determine the infinitesimal invariance conditions of infinitesimal generator representations of solution subgroups. There are many representations possible for spatial and dextrous manipulator homomorphisms. It turned out to be quite difficult to apply the procedures defined in (Olver, 1993) to determine solution groups of the dextrous manipulator homomorphism with a DH parameterisation of $\mathcal{F}$. It is difficult to derive the equation of the representation of the flow induced by the Jacobian of the homomorphism on SE(3) corresponding to the intrinsic flow of T^7 because of number of terms in the equation system. For an *n dof* manipulator, symbolic expansion of the first three rows of $\mathbf{T} = \prod_{j=1}^{n} \mathbf{T}_j$ produces a set of 12 equations in 18 unknowns. The 12 equations are highly nonlinear and contain hundreds of transcendental terms. To explicitly evaluate the vector field induced in the image by the homomorphism Jacobian involves taking the derivatives of these equations which generates a set of partial differential equations with even more terms.

While general algebraic routines are available to determine representations of solution groups in a given representation of sets of algebraic and differential equations, determination of the solution group representations by an infinitesimal generator analysis of the Lie algebra of infinitesimal generator of

field representations of solution subgroups, i.e. determination of G-invariance, would be intractable on a DH representation of the dextrous manipulator homomorphism. Hence an algebraic determination of dextrous manipulator solution groups was abandoned and instead a geometric factorisation was derived.

7.7 Geometric Factorisation of Canonical Solution Groups of a Homomorphism

Solution groups G_k^s of a homomorphism generated by manipulators are factored by a geometric analysis in Chapter 8 that is guided by an objective to maximise compactness of representation of solution group orbits and the solution element. The determination of G_k^{s*} and $\mathbf{y}_i^{s*}$ by geometric analysis whose representation is most compact aims to generate a canonical representation of G_k and $\mathbf{y}_i$ from the set of possible factorisations s of L and possible representations of these factorisations.

> Given a group of transformations acting on a L, by a canonical form of an element $\mathbf{y}_y \in \mathsf{L}$ is meant a distinguished, simple representative $\mathbf{y}_o$ of an the orbit, containing $\mathbf{y}_i$. The is no uniquely specified canonical form and some choice based on an aesthetic judgement of 'simplicity' is required. (Olver, 1995) pg. 42 ($\mathbf{y}_i$ notation substituted for the original).

Let $\mathsf{G}^s = \prod_{k=1}^{c} \mathsf{G}_k^s$ be a solution group factorisation that acts on $\mathbf{y}_i^s$ of L defined by $\mathcal{F}$. Let ${}^r\Psi_k^s, {}^r\Phi_k^s$ be representations of the G_k^s solution group operators that act on $\mathcal{F}$ pre-image ${}^r\boldsymbol{\theta}_i^s$ and image ${}^r\mathbf{x}_i^s$ representations respectively in atlas $A_r = \{U_\alpha^r, \phi_\alpha^r\}$. A canonical form representation of the solution element and the solution group actions is maximally representationally efficient, in a representation expression and transformation algorithm sense, in a given A_r.

Let L_1 be a measure of representational expression compactness of ${}^r\Psi_k^s, {}^r\Phi_k^s$. $L_1 = aL_1^m + bL_1^c$ is a weighted (by a, b) combined representation storage L_1^m and computational complexity L_1^c measure[10] of representational compactness of ${}^r\Psi_k^s, {}^r\Phi_k^s$.

Let L_2 be a measure of the compactness of representation of the solution element ${}^r\mathbf{y}_i^s$, generally a union of $o = 1, \ldots, z$ connected subsets of L. L_2 is generally a function L_3 of the summation of the areas L_3^o of each of these connected subsets. Each connected subset ${}^r_o\mathbf{y}_i^s$ has a boundary surface ${}^r\rho_o^s$ represented in atlas r of dimension one less than the dimension of ${}^r_o\mathbf{y}_i^s$. The hyper-surface area of ${}^r_o\mathbf{y}_i^s$ bounded by ${}^r\rho_o^s$ can be measured[11] by a surface integral over ${}^r\rho_o^s$ (Apostol, 1967). The L_3^o surface integral is usually well defined on subsets of L that do not contain $\mathbb{CP}^{\mathbb{S}}$. If ${}^r\rho_o^s = {}^rf_o^s(\phi_\alpha^r(\mathbf{a}))$ is parameterised by Euclidean chart variables $\mathbf{a}$ of dimension $\mathbb{R}^l$ in atlas $\{U_\alpha^r, \phi_\alpha^r\}$, then

$$L_3^o = \int_1 \cdots \int_l \| \prod_{h=1}^{l} \frac{\partial {}^rf_o^s}{\partial a_h} \| da_1 \ldots a_l \tag{7.24}$$

where $\prod$ is the vector cross product over the partial derivatives of ${}^rf_o^s$ with respect to the chart parameterisation coordinates. L_3^o can also be measured by

$$L_3^o = \int_1 \cdots \int_l \sqrt{\det \mathbf{B}} da_1 \ldots da_l \tag{7.25}$$

Δ

[10] For instance if ${}^r\Psi_k^s = \cos(\theta_i)$, $L({}^r\Psi_k^s) = a4 + b9$ if evaluation of $\cos(\theta_i)$ requires 9 transformation operations (assumed uniform cost) of θ and $\cos(\theta_i)$ is stored in a floating point precision that is four times more expensive than integer storage.

[11] A measure is a mapping (satisfying certain conditions) from certain subsets of a sigma algebra to a scalar field, usually the reals. A sigma algebra introduces additional closure and intersection of open and closed sets in the definition of a manifold. A measure is required in order to define an integral. A measurable manifold is integrable.

where $g_{cd} \in \mathbf{G} = \frac{\partial^r f_o^s}{\partial a_c} \cdot \frac{\partial^r f_o^s}{\partial a_d}$ and Δ is the restriction of the Euclidean chart parameterisation $\mathbf{a}$ to the ${}^r f_o^s$ pre-image of ${}^r \rho_o^s$. See (Osserman, 1986), pgs. 4-6. L_3^o evaluated by (7.24) or (7.25) does not depend on r since a surface integral is a coordinate system invariant measure. $L_3 = \sum_{o=1}^{z} L_3^o$. $L_2 = f_2(L_3)$. For instance, in the case of a nonparametric approximation of ${}^r\mathbf{y}_i^s$, the compactness of the approximation L_3 will be a function of the surface area L_2 of ${}^r\mathbf{y}_i^s$. The geometric factorisation of L in Chapter 8 to derive a canonical form representation of the ${}^{r*}\mathbf{y}_i^{s*}$ and ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ attempts to determine $(s*, r*)$ that minimise the canonical solution group representation objective function

$$L_{\min} = \min_{s,r}[aL_2({}^r\mathbf{y}_i^s) + b \sum_{k=1}^{c=f(s)} L_1({}^r\Psi_k^s) + L_1({}^r\Phi_k^s)]. \tag{7.26}$$

where a and b weight the importance of representation compactness of the solution element vs the solution group operators. s^* is the most compact factorisation of L that has the most compact representation in representation $r*$. (7.26) is a difficult function to minimise. Since automatic procedures for generating manifold atlases and combinatorial search of solution group factorisations could not be derived, in Chapter 8 a geometric analysis is used to minimise (7.26). Since a geometric analysis is employed, the minimisation of (7.26) is not guaranteed to be the unique[12] minimum solution group and its representation, only a good minimum.

The criteria for determination of $(s*, r*)$ are that:

1. ${}^r\Psi_k^s, {}^r\Phi_k^s$ representation of G_k^s act component-wise independently on $\boldsymbol{\theta}_i$, $\mathbf{x}_i$ respectively.
2. ${}^r\Psi_k^s$ or ${}^r\Phi_k^s$ act only on the pre-image and the image respectively of $\mathcal{F}$.
3. L_2 is minimal.

The more factors G_k^s that are derived in a given G^s factorisation, the smaller is L_2 of $\mathbf{y}_i^s$ on which G_k^s act. However a complete factorisation of L so that $\mathbf{y}_i^s$ is a point may result in very complicated expressions for ${}^r\Psi_k^s, {}^r\Phi_k^s$. (7.26) is minimised by a maximal factorisation of L that has the most compact representation ${}^r\Psi_k^s, {}^r\Phi_k^s$ and a maximally compact representation of $\mathbf{y}_i^s$ that is acted on by ${}^r\Psi_k^s, {}^r\Phi_k^s$ solution group operators. A factorisation of G_k^s actions on L which are independent in the coordinates of the representation of pre-image and image variables generally leads to efficient representations of ${}^r\Psi_k^s, {}^r\Phi_k^s$ actions.

[12] There may be also be solution groups over s, r of (7.26).

Chapter 8

Canonical Solution Groups of a Homomorphism

Redundant *dof* enable a manipulator and a human to perform $\boldsymbol{\theta}_i$ self-motions while maintaining position and orientation fixed in order to maximise multiple inverse solution optimality criteria such as manipulability and obstacle avoidance. Iterative differential, Lyapunov and optimisation methods for inversion of a multi-branch infinite solution inverse mapping may be slow and acyclic. This Chapter derives canonical solution groups of kinematic mappings of dextrous manipulators and a local linear approximation of the solution element. The exact solution group operators act on the solution element representation to generate an approximation of the inverse kinematic solution groups of a manipulator. This inverse kinematic solution method is cyclic and would be fast if implemented on parallel hardware. The solution is proposed as a model of sensorimotor coordination in primates in Chapter 9.

8.1 Approximation of the Canonical Solution Groups of Dextrous Manipulators

From the previous Chapter, $\mathsf{G}^s = \prod_{k=1}^{c} \mathsf{G}_k^s$ is a solution group factorisation that acts on $\mathbf{y}_i^s$ of L defined by $\mathcal{F}$ and ${}^r\Psi_k^s, {}^r\Phi_k^s$ are representations of the G_k^s solution group operators that act on $\mathcal{F}$ pre-image ${}^r\boldsymbol{\theta}_i^s$ and image ${}^r\mathbf{x}_i^s$ representations respectively in atlas $A_r = \{U_\alpha^r, \phi_\alpha^r\}$. A canonical form representation defined by s^*, $r*$ of the solution element and the solution group actions is maximally representationally effecient, in a representation expression and transformation algorithm sense, in a given A_r. The criteria for determination of $(s*, r*)$ are that:

1. ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ representation of G_k^{s*} act component-wise independently on $\boldsymbol{\theta}_i$, $\mathbf{x}_i$ respectively,
2. and L_2 is minimal.

A canonical factorisation represented by ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ is a maximal factorisation of L that has the most compact representation ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ and a maximally compact representation of $\mathbf{y}_i^{s*}$ that is acted on by ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ solution group operators. A factorisation of G_k^{s*} actions on L which are independent in the coordinates of the representation of pre-image and image variables generally leads to effecient representations of ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ actions.

After $\mathbf{y}_i^{s*} = \boldsymbol{\theta}_i^{s*}$ is determined by geometric analysis, it must also be represented. $\mathbf{y}_i^{s*}$ is generally a set of trivial fibre bundles. A trivial fibre bundle may be parameterised by a canonical[1] Γ_w or a direct inverse function Υ_w^{-1} coordinatisation

$$\Upsilon_w^{-1} : \boldsymbol{\theta}_i^{s*} = \Upsilon_w^{-1}(\mathbf{x}_i^{s*}, \zeta), \ \forall \mathbf{x}_i^{s*} \in \mathfrak{W}_w, \tag{8.1}$$

where $\zeta = f(\boldsymbol{\theta}_i^{s*})$ is a direct parameterisation of the solution element fibre $\boldsymbol{\theta}_i^{s*}$ at $\mathbf{x}_i^{s*} \in \mathfrak{W}_w$. See Definition 7.13. Υ_w^{-1} are deterministic smooth functions that are guaranteed to be locally cyclic within $\mathfrak{W}_w$ (Wampler, 1988). Υ^{-1} can be solved in closed form or approximated[2] by F^{-1}, an interpolant or a function approximator. If F^{-1} is an approximation of Υ^{-1}, the action of the exact canonical solution operators ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ on $\boldsymbol{\theta}_i^{s*} = \Upsilon_w^{-1}(\mathbf{x}_i^{s*}, \zeta)$ generates an approximate action of the canonical solution group of $\mathcal{F}$.

8.1 Definition (The Actions of the Canonical Solution Group on the Solution Element)
Given ${}^{r*}\Psi_k^{s*}(\boldsymbol{\theta}_i^{s*}) = F_w^{-1}({}^{r*}\Phi_k^{s*}(\mathbf{x}_i^{s*}), \zeta)$, if $\mathbf{x}_j^{s*} \notin \mathbf{x}_i^{s*}$ is an inverse solution query, in order to determine $\boldsymbol{\theta}_j^{s*}$ using the solution group approximation, it is necessary to determine the solution group operator ${}_j^{r*}\Phi_1^{s*}, \ldots, {}_j^{r*}\Phi_c^{s*}$ transformations of the image of $\mathcal{F}$ such that

$$(\mathbf{x}_i^{s*}) = {}_j^{r*}\Phi_c^{s*}(\ldots {}_j^{r*}\Phi_1^{s*}(\mathbf{x}_j^{s*})\ldots), \tag{8.2}$$

and the corresponding inverse transformations ${}_j^{r*}\Psi_1^{s*^{-1}}, \ldots, {}_j^{r*}\Psi_c^{s*^{-1}}$ of the pre-image of $\mathcal{F}$ such that

$$\boldsymbol{\theta}_j^{s*} = {}_j^{r*}\Psi_1^{s*^{-1}}(\ldots {}_j^{r*}\Psi_c^{s*^{-1}}(F_w^{-1}({}_j^{r*}\Phi_c^{s*}(\ldots {}_j^{r*}\Phi_1^{s*}(\mathbf{x}_j^{s*})\ldots), \zeta))\ldots). \tag{8.3}$$

8.1.1 Geometric Factorisation of the Canonical Solution Groups

The main criterion for geometric factorisation G_k^{s*} of L is that ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ representation of G_k^{s*} act component-wise independently on $\boldsymbol{\theta}_i$, $\mathbf{x}_i$ respectively. For manipulator kinematic homomorphisms, there are basically two types of transformations that have this property:

1. $\Phi_k^{s*} = \mathbf{I}$, $\mathbf{I}$ an identify transfomation and ${}^{r*}\Psi_k^{s*}$ a representation of $\mathsf{C}_2^k, k = 1, \ldots 4$, a two cyclic solution branch transformation. Four two cyclic solution group operators generate 16 possible solution branches, the maximum for a manipulator. See (Burdick, 1988).

2. ${}^{r*}\Psi_k^{s*} = f_1(\urcorner_k), {}^{r*}\Phi_k^{s*} = f_2(\urcorner_k)$, $\urcorner_k$ a component of the pre-image and $\urcorner_k$ also a coordinate function of the image of $\mathcal{F}$ in representation $r*$.

Determination of r^* and s^* such that the number of k which satisfy the second property is maximised is the principle objective of a geometric factorisation of the solution groups. This is achieved by definition of

8.1 Lemma ($\mathbb{R}^n\backslash 0$ can be Factored by $\mathbb{R}^+ \times \mathsf{S}^{n-1}$)
$(\mathbb{R}^n\backslash 0, +)$, an n-dimensional real continuum minus the n-dimensional origin, i.e. a truncated[3] group, may be factored by S^{n-1}. $\mathbb{R}^n\backslash 0 = \mathbb{R}^+ \times \mathsf{S}^{n-1}$, where $\mathbb{R}^+$ are the positive reals excluding 0. A chart $f : \mathsf{S}^{n-1} \times \mathbb{R}\backslash 0 \rightarrow \mathbb{R}^n$ is defined everywhere except at 0. f defines a spherical coordinate chart of $\mathbb{R}^n$ defined everywhere except at the origin. A complete spherical atlas of $\mathbb{R}^n$ requires two overlapping charts f_1, f_2.

[1] See §5.3.4.
[2] F^{-1} is not an inverse of an approximation function F. The inverse function notation denotes that it is an approximation of a direct inverse function.
[3] $(\mathbb{R}^n, +)$ cannot be factored by S^1 because there is an origin singularity of $f\mathsf{S}^{n-1} \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$.

As special cases of Lemma 8.1, $\mathbb{R}^2\backslash(0,0) = \mathbb{R}^+ \times \mathsf{S}^1$, $\mathbb{R}^3\backslash(0,0,0) = \mathbb{R}^+ \times \mathsf{S}^2$. $\mathsf{SO}(n)\backslash\mathbf{a}, \mathbf{a}$ any rotation, can be represented by $\mathbf{S}^n$. See §7.4. Hence $\mathsf{SO}(n)\backslash\mathbf{a}$ can be factored by $\mathbf{S}^m, m < n$. T^n can be factored by S by definition. See §7.4. It is a little difficult to geometrically determine ${}^{r*}\Psi_k^{s*} = f_1(\urcorner_k), {}^{r*}\Phi_k^{s*} = f_2(\urcorner_k)$ for 7 *dof* manipulators because $\mathcal{F}$ couples the position and orientation of the manipulator in SE(3). See footnote 20 on pg. 199.

8.1.2 The Homotopy of the Trivial Fibre Bundle Solution Element

After ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$ have been derived by geometric analysis, they act on the solution element $({}^r\boldsymbol{\theta}_i^s, {}^r\mathbf{x}_i^s)$. If the solution group factorisation of a dextrous manipulator is defined on a single point of the base space, the solution element is a fibre $\mathfrak{F}$. For incomplete factorisation of the base space, the solution element has the structure of a fibre bundle. The global fibre bundle solution on ${}^r\mathfrak{W}$ is a union of trivial fibre bundle solutions on ${}^r\mathfrak{W}_w, w = 1, \ldots, r$. Each ${}^r\mathfrak{W}_w$ is a maximally connected subset of ${}^r\mathfrak{W}_w$ without $\mathbb{CPS}$. See Definition 2.7. ${}^r\mathfrak{W}_w$ are partitioned by $\mathbb{CPS}$ of $\mathcal{F}$.

8.2 Lemma (A Kinematic Homeomorphism Solution Element is Locally a Trivial Fibre Bundle) Let ${}^r\mathfrak{W}_w$ be maximal $\mathbb{CPS}$-free regions of ${}^r\mathfrak{W} \subset \mathsf{C}^n$. Let ${}^r\boldsymbol{\theta}_i^s \subset {}^r\ \mathsf{T}^m$ be a connected region such that $\Upsilon_i^{-1}({}^r\boldsymbol{\theta}_i^s) = {}^r\ \mathfrak{W}_i$. $\forall {}^r\mathbf{x}_i^s \in {}^r\ \mathfrak{W}_i, \Upsilon_i^{-1}({}^r\mathbf{x}_i^s, \zeta)$ is diffeomorphic to T^{m-n} (DeMers, 1993). See Theorem 2.1.

The topological structure of $\Upsilon_i^{-1}({}^r\mathbf{x}_i^s, \zeta)$ can only change at $\mathbb{CPS}$. Hence $\Upsilon_i^{-1}({}^r\mathbf{x}_i^s, \zeta), \mathbf{x}_i \in {}^r\ \mathfrak{W}_i$ is diffeomorphic to $\Upsilon_i^{-1}(\mathbf{x}_j, \zeta), \mathbf{x}_j \in {}^r\ \mathfrak{W}_i$. For complete solution group factorisation of the base space of a dextrous manipulator, the pre-image fibre $\mathfrak{F}$ of a regular point solution element component ${}^r\mathbf{x}_i^s \in {}^r\ \mathfrak{W}_i$, the workspace of a dextrous manipulator defined by (2.7), is homeomorpic to T^{n-m}. For incomplete solution group factorisation of the base space, the solution element is canonically a product of a collection of compact subsets of ${}^r\mathfrak{W}_i, i = 1, \ldots, d$, and the associated canonical T^{n-m} pre-image fibres, i.e. a collection of trivial fibre bundles. The canonical T^{n-m} solution $\mathfrak{F}$ may have different homotopy classes in different ${}^r\mathfrak{W}_i$ of the collection. See (Baker, 1990; Burdick, 1988). In this case, the collection of trivial fibre bundle solutions is not trivial[4] and it does not have a surjective inverse function. Solution element $\mathfrak{F}$ homotopy class changes as it crosses $\mathbb{CPS}$ between trivial fibre bundles defined on sub-workspaces ${}^r\mathfrak{W}_i$.

There are two solution $\mathfrak{F}$ homotopies - contractable or non-contractable. $\mathfrak{F}$ is possibly in l disjoint pieces $\mathfrak{F}_1, \ldots, \mathfrak{F}_l$ corresponding to l solution branches. See Definition 7.14. Each $\mathfrak{F}_1, \ldots, \mathfrak{F}_l$ may be non-contractable along a given $\mathcal{F}$ pre-image coordinate. If the solution element spans ${}^r\mathfrak{W}_i, {}^r\ \mathfrak{W}_j$ where ${}^r\mathfrak{W}_i$ and ${}^r\mathfrak{W}_j$ are maximally-connected $\mathbb{CPS}$-free subspaces of ${}^r\mathfrak{W}$ and ${}^r\mathfrak{W}_i \cap {}^r\ \mathfrak{W}_j = \emptyset$, the pre-image fibres of different points ${}^r\mathbf{x}_i^s \in {}^r\ \mathfrak{W}_i, {}^r\mathbf{x}_j^s \in {}^r\ \mathfrak{W}_j$ may belong to different homotopy classes. $\mathfrak{F}$ may change homotopy from connected to disjoint or from single disjoint to multiple disjoint moving from ${}^r\mathfrak{W}_i$ to ${}^r\mathfrak{W}_j$. The global fibre bundle structure of $\mathcal{F}$ is defined by these changes in homotopy, i.e. solution branch structure.

The homotopy structure of the collection of trivial fibre bundle solution elements impacts the solution group cyclic branch factorisation $\Phi_k^{s*} = \mathbf{I}, {}^{r*}\Psi_k^{s*}$ a representation of $\mathsf{C}_2^k, k = 1, \ldots 4$. For solution elements ${}^r\mathbf{x}_i^s \in {}^r\ \mathfrak{W}_i$ with a contractable fibre $\mathfrak{F}_i$ and ${}^r\mathbf{x}_j^s \in {}^r\ \mathfrak{W}_j$ with a noncontractable $\mathfrak{F}_j$ fibre in $\mathfrak{F}_j^1, \ldots, \mathfrak{F}_j^l; l = 2, 4, 8, 16$, disjoint pieces, $\mathsf{C}_2^k, k = 1, \ldots, 4$ are the corresponding solution branch factors, represented by ${}_i^{r*}\Psi_k^{s*}$, ${}_j^{r*}\Psi_k^{s*}$ respectively, of both $\mathfrak{F}_i$ and $\mathfrak{F}_j$. Strictly speaking, see Definition 7.14, ${}_i^{r*}\Psi_k^{s*}$ are not branch factors since $\mathfrak{F}_i$ is contractable but a ${}_i^{r*}\Psi_k^{s*}$ factorisation of

[4] Neither is it non-trivial, for instance a Möbius strip. This is homotopic to $\mathsf{S}^1 \times \mathsf{I}$ with a I fibre orienation reversal at one point. A non-trivial S^1 fibre bundle is minimally embedded in four dimensions, for instance a Klein bottle. See (Weyl, 1952).

$\mathfrak{F}_i$ may be straightforward to derive and $\mathfrak{F}_i$ may be sectioned in a manner reminiscent[5] of a solution branch. For ${}^r\mathbf{x}_j^s \in^r \mathfrak{W}_j$, $\Upsilon_j^{-1}({}^r\mathbf{x}_j^s, \zeta)$ parameterises $\mathfrak{F}_j$. If ${}_j^{r*}\Psi_k^{s*}$ a representation of $\mathsf{C}_2^k, k = 1, \ldots 4$ have been derived, they act on $\Upsilon_j^{-1}({}^r\mathbf{x}_j^s, \zeta_l)$, where ζ_l is a restricted parameterisation of $\mathfrak{F}_j$ paramaterised by restricted Υ_j^{-1} to a single $\mathfrak{F}_j$, in the following manner

$$ {}^r_{1234}\boldsymbol{\theta}_j^s = {}_j^{r*}\ \Psi_4^{s*}(\ldots {}_j^{r*}\ \Psi_1^{s*}(\Upsilon_j^{-1}({}^r\mathbf{x}_j^s, \zeta_l))\ldots) \tag{8.4} $$

The presence of a digit in the prefix subscript 1234 of ${}^r_{1234}\boldsymbol{\theta}_j^s$ represents the presence of the corresponding digit k indexed cyclic factor in (8.4), i.e. the prefix digits indexes the cyclic branch solution group products.

8.1.3 Approximation of the Solution Element

If Υ^{-1} cannot be represented in closed-form, it can be approximated by F_j^{-1}. For dextrous manipulators, the solution fibre $\mathfrak{F}$ is homotopic to T^{n-m} and is embedded in a pre-image homotopic to T^n. If $n - m = \dim(\zeta)$, since a representation of T^{n-m} is a multi-valued function[6], a single valued $\Upsilon_j^{-1}({}^r\mathbf{x}_j^s, \zeta_{\leftarrow})$ is defined[7] for a restriction $\leftrightarrow$ of ζ to a paramaterisation of $\frac{\mathfrak{F}}{2^{n-m}}$ where $\frac{\mathfrak{F}}{2^{n-m}}$ is a 2^{n-m}-ant[8] defined by $n - m$ bisections of $\mathfrak{F}$ by its $n - m$ canonical axes.

For non-contractable $\mathfrak{F}$ homotopic to T^{n-m}, the determination of the subset of $\mathfrak{F}$ over the base subspaces on which Υ^{-1} is defined is related to the determination of the k solution group branch factors C_2^k of the l solution branches as shown by (8.4). If $n - m = 1$, i.e. $\mathfrak{F}$ is homotopic to S^1 and if there are only two solution branches C_2^1, e.g. planar 3-$\mathbf{R}$, then these solution branches are a bisection of $\mathfrak{F}$ into two $\frac{\mathfrak{F}}{2}$. Υ^{-1} and hence F^{-1} can be restricted to either of these by a ζ_2 parameterisation of dimension one. In the case of $\mathfrak{F}$ homotopic to S^1 and $l > 2$, i.e. 4,8, and 16 solution branches, then the corresponding Υ^{-1} is defined on ζ_l restricted to any of the $\frac{\mathfrak{F}}{l}$ pieces.

The F^{-1} approximations of Υ^{-1} of 3 and 7 *dof* dextrous manipulators proposed in this Chapter are defined on $\zeta_{\leftarrow}$ parameterisations (of dimension $n - m$) restricted to $\frac{\mathfrak{F}}{2^{n-m}}$ for contractable $\mathfrak{F}$ and ζ_l (of dimension $n - m$) restricted to $\frac{\mathfrak{F}}{l}$ if $l > 2^{n-m}$ or[9] restricted to $\frac{\mathfrak{F}}{2^{n-m}}$ if $l < 2^{n-m}$ for disjoint non-contractable $\mathfrak{F}$.

The determination of a restriction of $\zeta_{\leftarrow}$ to $\frac{\mathfrak{F}}{2^{n-m}}$ or ζ_l to $\frac{\mathfrak{F}}{l}$ over a given point ${}^r\mathbf{x}_i^s \in \mathfrak{W}_w$ is a little complicated since while $\mathfrak{F}$ is homotopic to T^{n-m}, $\mathfrak{F}$ may not be an exact $n - m$-torus embedded in T^n. $\mathfrak{F}$ above a given base space $\mathfrak{W}_w$ expands, contracts, warps and changes its orientation relative to the coordinate axes of the representation of T^n, over different points in $\mathfrak{W}_w$. The determination of the $\theta_{m+1}, \ldots, \theta_n \subset^r \boldsymbol{\theta}_i^s$ such that $\theta^{\zeta} = \theta_{m+1}, \ldots, \theta_n$ depends on the homotopy analysis of each $\mathfrak{W}_w$. For a disjoint $\mathfrak{F}, \theta_{m+1}, \ldots, \theta_n$ are determined by those dimensions of the embedding space homotopic to T^n along which $\mathfrak{F}$ is non-contractable and not those dimensions along which the fibre is disjoint since this is a dimension where a restriction of ζ_l is generated. For contractable $\mathfrak{F}, \theta_{m+1}, \ldots, \theta_n$ should be selected as those dimensions of the embedding space homotopic to T^n in which $\mathfrak{F}$ has the largest dynamic range since this makes the restriction of $\zeta_{\leftarrow}$ easier to determine.

There is a further complication in determination of the restriction of $\zeta_{\leftarrow}$ over $\mathfrak{W}_1$, the workspace between the workspace envelope $\mathbb{CP}\mathbb{S}_0$ and the $\mathbb{CP}\mathbb{S}_1$ closest to $\mathbb{CP}\mathbb{S}_0$. $\mathfrak{F}$ over $\mathfrak{W}_1$ is generally of

[5] For instance, elbow up, elbow down solution sections of $\mathfrak{F}_i$. These are not strictly branches since $\mathfrak{F}_i$ is contractable and hence ζ^g exists to convert one solution to another.

[6] For instance The circle S^1 has two ordinate values for each abscissa value.

[7] Alternativly, ζ must be extended by additional variables, for instance an integer variable h, that index the multi-valued solutions for the unextended ζ of dimension $n - m$ in each of the multi-valued solution fibre subsurfaces $\mathfrak{F}_h, h = 1, \ldots, 2^{n-m}$.

[8] For instance a quadrant, or an three dimensional octant of a Torus.

[9] Since this case does not actually arise, analysis of the restriction of $\zeta_{\leftarrow}$ to subsets of $l > 2$ branches of solution fibres homotoptic to $T^{n-m>1}$ is not pursued further.

contractable homotopy. However the actual fibre is incomplete[10]. The solution fibre at $\mathbb{CP}\mathbb{S}_0$ is generally a point. As $\mathcal{F}$ moves away from $\mathbb{CP}\mathbb{S}_0$ into $\mathfrak{W}_1$, $\mathfrak{F}$ remains contractable to a point, i.e. any fibre solution can be converted to another solution by a fibre traversal. However the traversal is restricted to a subset of directions along the fibre. For instance when $\mathfrak{F}$ is homotopic to S^1, conversion of one solution to another by fibre traversal is only defined in one traversal direction along a fibre that looks like a 'smile' when plotted in its embedding space. For $\mathfrak{F}$ homotopic to S^1, embedded in T^{n-m}, $\mathfrak{F}$ grows from a point at $\mathbb{CP}\mathbb{S}_0$, to a steadily enlarging 'smile' which becomes a circle at $\mathbb{CP}\mathbb{S}_1$ and forks into l disjoint non-contractable fibres in $\mathfrak{W}_2$, where the fibres remain homotopic to S^1. In general Υ^{-1} will be defined on sets of $\mathfrak{W}_w$ under trivial contractable, trivial contractable incomplete, and trival non-contractable disjoint fibres.

$\mathcal{F}$, and its Jacobian $\mathbf{J}$ and Hessian $\mathbf{H}$ can be evaluated. If $\mathcal{F}$ is augmented by $n-m$ direct coordinatisation functions $\zeta^1 = f_{m+1}, \ldots, \zeta^{n-m} = f_n$, then the corresponding augmented Jacobian $\mathbf{J}^{\Diamond}$ (see §2.6.1) and augmented Hessian $\mathbf{H}^{\Diamond}$ can be evaluated. With a careful choice of ζ, $\mathbf{J}^{\Diamond^{-1}}$ and $\mathbf{H}^{\Diamond^{-1}}$ can be evaluated. $\mathbf{J}^{\Diamond^{-1}}$ and $\mathbf{H}^{\Diamond^{-1}}$ are the inverse augmented Jacobian and the inverse augmented Hessian[11] of Υ^{-1} and hence of $\mathcal{F}^{-1}$. $\mathcal{F}^{-1}$ can be constructed on a tessellation of ${}^r\mathbf{x}_i^s$ times a tessellation of $\zeta_{\leftharpoondown}$ which induces a restricted tesselation of ${}^r\boldsymbol{\theta}_i^s$.

8.1.3.1 A Restricted Tessellation of the Solution Element

Let $\aleph^d$ be a tessellation operator that generates a d density regular tessellation of a Euclidean or Riemannian metric space. $\aleph^d$ maps a bounded hyper-surface to a set of point elements of a bounded Euclidean or Riemannian surface. Examples of $\aleph^d$ are orthogonal, geodesic and Delaunay tessellation operators (see §9.4). ${}^r_{\aleph}\mathbf{x}_i^s = \aleph^d({}^r\mathbf{x}_i^s)$, ${}_{\aleph}\zeta_{\leftharpoondown} = \aleph^d(\zeta_{\leftharpoondown})$ and ${}^r_{\aleph}\boldsymbol{\theta}_i^s = \aleph^d({}^r\boldsymbol{\theta}_i^s)$. A regular ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$ tesselation induces an irregular $\sharp$ tesselation ${}^r_{\sharp}\boldsymbol{\theta}_i^s$ of the image of Υ^{-1} by ${}^r_{\sharp}\boldsymbol{\theta}_i^s = \Upsilon^{-1}({}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown})$ where Υ^{-1} is determined by differential inversion of $\mathbf{J}^{\Diamond^{-1}}$. A regular ${}^r_{\aleph}\boldsymbol{\theta}_i^s$ induces an irregular tesselation ${}^r_{\sharp}\mathbf{x}_i^s \times_{\sharp} \zeta_{\leftharpoondown} = \Upsilon({}^r_{\aleph}\boldsymbol{\theta}_i^s)$ where $\Upsilon = \mathcal{F}$. A direct regular ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$ is the best tessellation strategy for construction of $\mathcal{F}^{-1}$. $\mathcal{F}^{-1}$ approximation error is more uniform and error upper bounds are easier to analyse when $\mathcal{F}^{-1}$ is constructed on a direct regular ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$ instead of ${}^r_{\aleph}\boldsymbol{\theta}_i^s$. A direct regular orthogonal and geodesic tessellation of ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$ is a uniform tessellation of $\mathcal{F}$ curvature and hence of the curvature induced by $\mathcal{F}^{-1}$ in the solution element fibre bundle. $\mathcal{F}$ average and maximum approximation error are proportional to the tessellation density d of ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$.

Direct ${}^r_{\aleph}\mathbf{x}_i^s \times_{\aleph} \zeta_{\leftharpoondown}$ combined with differential inversion of an augmented Jacobian can generate a restricted $\zeta_{\leftharpoondown}$ or ζ_l parameterisation of the restricted $\frac{\mathfrak{F}}{2^{n-m}}$ or $\frac{\mathfrak{F}}{l}$ respectively. In the case of a $\mathcal{F}$ approximation of a solution element fibre bundle with solution fibre $\mathfrak{F}$ homotopic to S^1 examined in the rest of the Chapter, once θ^{ζ}, that dimension of the embedding space T^3 or T^7 of $\mathfrak{F}$ that parameterises $\mathfrak{F}$ has been determined by homotopy analysis, θ^{ζ} can be restricted $\theta^{\zeta\leftharpoondown}$ to $\frac{\mathfrak{F}}{2}$ over the base space of a disjoint fibre by combining differential $\mathbf{J}^{\Diamond^{-1}}$ solution inversion with check of stored solutions for tessellation neighbours in ${}^r_{\aleph}\mathbf{x}_i^s$ in the case of disjoint $\mathfrak{F}$ and with check of stored solutions along $\mathfrak{F}$ for a single point in the case of contractable $\mathfrak{F}$.

Let ${}^1\theta^{\zeta\leftharpoondown}$ be a first 1 value of $\zeta_{\leftharpoondown}$ for a given a point ${}^r\mathbf{x}_i^s \in \mathfrak{W}_w$, a solution element subspace of $\mathfrak{W}_w$. After selection of arbitrary ${}^r_0\boldsymbol{\theta}_i^s$, differential inversion of $({}^r\mathbf{x}_i^s, {}^1\theta^{\zeta\leftharpoondown})$ by $\mathbf{J}^{\Diamond^{-1}}$ will generate solution ${}^r_1\boldsymbol{\theta}_i^s$. Differential inversion of $({}^r\mathbf{x}_i^s, {}^q\theta^{\zeta\leftharpoondown} = {}^1\theta^{\zeta\leftharpoondown} + q\Delta)$ where Δ is the tessellation edge length of ${}_{\aleph}\zeta_{\leftharpoondown}$ by

[10] A fibre plot shows a torus with a chunk out of it.

[11] $\mathbf{H}^{\Diamond^{-1}}$ measures the curvature of Υ^{-1}. The eigenvectors $\mathbf{w}_{\nu_e}$ of $\mathbf{H}^{\Diamond^{-1}}$ represent the principle directions of curvature. The magnitude of curvature along the direction of a given eigenvector $\mathbf{w}_{\nu_e}$ is determined by the magnitude of the eigenvalue ν_e along that direction. Eigenvalues of $\mathbf{H}^{\Diamond^{-1}}$ over the pre-image of Υ^{-1} are a continuous function of the pre-image.

$\mathbf{J}^{\Diamond^{-1}}$ will generate a q solution ${}^{r}_{q}\theta^{s}_{i}$ local to ${}^{r}_{q-1}\theta^{s}_{i}$ for ${}^{r}_{q}\theta^{s}_{i}$ if $\mathbf{J}^{\Diamond^{-1}}$ is initalised at ${}^{r}_{q-1}\theta^{s}_{i}$.

For disjoint fibres, $\forall q, {}^{r}_{q}\theta^{s}_{i}$ are guaranteed to be restricted to the solution branch $\frac{\mathfrak{F}}{2}$. For a disjoint $\mathfrak{F}, \theta^{\zeta-}$ is determined by homotopy analysis as that dimension of the embedding space homotopic to Υ^{n} along which $\mathfrak{F}$ is non-contractable. Hence the restriction of $\theta^{\zeta-}$ is defined for $\theta^{\zeta-} = 0, \ldots, z\Delta = 2\pi$ where z is the number of tessellation points of ${}_{\sharp}\theta^{\zeta-}$.

If homotopy analysis fails to determine the homotopy of the fibre over $\mathfrak{W}_{w}$, a search over ${}_{d}\theta^{\zeta-}$ parameterisations for each dimension d of the embedding space homotopic to $\mathfrak{T}^{n}$ along which $\mathfrak{F}$ is non-contractable can determine fibre homotopy. For each dimension d, where ${}_{d}\theta^{\zeta-}$ is the parameterisation of the fibre, after selection of arbitrary ${}^{r}_{0}\theta^{s}_{i}$, differential inversion of $({}^{r}\mathbf{x}^{s}_{i}, {}^{q}_{d}\theta^{\zeta-} =^{1}_{d} \theta^{\zeta-} + q\Delta)$ by $\mathbf{J}^{\Diamond^{-1}}$ will generate a q solution ${}^{r}_{q}\theta^{s}_{i}$. If ${}^{r}_{q}\theta^{s}_{i}$ are defined for ${}_{d}\theta^{\zeta-} = 0, \ldots, 2\pi$ for ${}_{d}\theta^{\zeta-}$ in dimension d of the embedding space, then the fibre is non-contractable over dimension d, i.e. it has disjoint homotopy. If ${}^{r}_{q}\theta^{s}_{i}$ are not defined for ${}_{d}\theta^{\zeta-} = 0, \ldots, 2\pi$ for ${}_{d}\theta^{\zeta-}$ for any d, then the fibre is contactable.

For a disjoint fibre, the dimension d of the embedding space where the solution branch fibres are disjoint can be determined by a combinatoric search of the initialisations $\{{}^{r}_{0}\theta^{s}_{i}\}$ of $\mathbf{J}^{\Diamond^{-1}}$ inversion of $({}^{r}\mathbf{x}^{s}_{i}, {}^{1}\theta^{\zeta-})$. $\{{}^{r}_{0}\theta^{s}_{i}\}$ can be generated by adding π to each of the coordinates of ${}^{r}_{1}\theta^{s}_{i}$. Dimension d is the disjoint fibre seperation dimension if the $\mathbf{J}^{\Diamond^{-1}}$ solution converges to ${}^{r}_{1}\theta^{s}_{i_d} \neq^{r}_{1} \theta^{s}_{i}$.

For contractable fibres, generation of the restriction of $\digamma$ to $\frac{\mathfrak{F}}{2}$ is a little more complicated. $\mathfrak{F}$ expands, contracts, warps, and changes its orientation, relative to the coordinate axes of the representation of the embedding space Υ^{n}, over the solution element subspace of $\mathfrak{W}_{w}$. A restriction of $\theta^{\zeta-}$ will have variable range over $\mathfrak{W}_{w}$. To determine the end-points in the embedding space coordinates of given semicircular fibre restriction over $\mathfrak{W}_{w}$, it is necessary to monitor the sign of the coordinate d differences sign_{d} in solutions generated by $\mathbf{J}^{\Diamond^{-1}}$ traversal along $({}^{r}\mathbf{x}^{s}_{i}, {}^{q}\theta^{\zeta-} =^{1} \theta^{\zeta-} + q\Delta)$ as q increases. If for any d

$$\text{sign}_{d}({}^{r}_{q+1}\theta^{s}_{i} -^{r}_{q} \theta^{s}_{i}) \neq \text{sign}_{d}({}^{r}_{q}\theta^{s}_{i} -^{r}_{q-1} \theta^{s}_{i}) \tag{8.5}$$

a semifibre endpoint exists between solutions q and $q+1$. The end point may be located accurately over ${}^{1}\theta^{\zeta-} + u\Delta$, $u \in \mathbb{R}$ by for instance a golden section search. See (Press et al., 1992). Once the other end point of the semicircle has been located accurately in a similar manner at ${}^{1}\theta^{\zeta-} - v\Delta$ for solutions generated by $\mathbf{J}^{\Diamond^{-1}}$ traversal q along $({}^{r}\mathbf{x}^{s}_{i}, {}^{-q}\theta^{\zeta-} =^{1} \theta^{\zeta-} - q\Delta)$, the range of $\theta^{\zeta-}$ to parameterise the semifibre over ${}^{r}\mathbf{x}^{s}_{i}$ is $u+v$. The restriction of $\theta^{\zeta-}$ is defined for ${}^{q}\theta^{\zeta-} =^{1} \theta^{\zeta-} - v\Delta + q\frac{u+v}{z}$ where z is the number of tessellation points of ${}_{\aleph}\theta^{\zeta-}$.

For disjoint fibres, for restriction of $\theta^{\zeta-}$ to $\frac{\mathfrak{F}}{2}$ over the complete solution element subspace of $\mathfrak{W}_{w}$, the solution ${}^{r}_{1}\theta^{s}_{j}$ for a nearest neighbour point j inverted for $({}^{r}\mathbf{x}^{s}_{j}, {}^{1}\theta^{\zeta-})$ by $\mathbf{J}^{\Diamond^{-1}}$ will be restricted to $\frac{\mathfrak{F}}{2}$ if $\mathbf{J}^{\Diamond^{-1}}$ is initalised at ${}^{r}_{1}\theta^{s}_{i}$.

The application of the above procedures to ${}^{r}_{\aleph}\mathbf{x}^{s}_{i} \times_{\aleph} \zeta_{-}$ induces a restricted tessellation ${}^{r}_{\sharp}\theta^{\zeta-}_{i}$ of the Υ^{-1} image. The same procedures may be applied to generation of ${}^{r}_{\sharp}\theta^{\zeta^{l}-}_{i}$ over ${}^{r}_{\aleph}\mathbf{x}^{s}_{i} \times_{\aleph} \zeta_{-}$ for the other solution branch l fibres or semicircular fibre with the simplification that semicircular fibre endpoints are the same as those calulated for the first semicircular fibre if these have been stored.

8.1.3.2 A Local Linear Approximation of the Solution Element

A local linear approximation of Υ^{-1} is proposed in this section since an exact parameterisation of the solution group element of certain classes of dextrous manipulators could not be derived in closed-form. Smooth functions such as Υ^{-1} can be approximated by polynomial expansions such as a Taylor approximation. See (Apostol, 1967). In general an n^{th} order Taylor series expansion is exact for n^{th} order polynomial approximation. While $\mathcal{F}$ is continuous and differentiable, since $\mathcal{F}$ is a composition

of transcendental functions, in general n^{th} order approximation will be inexact, i.e. $R_n \neq 0$. The radius of convergence of an infinite series Taylor approximation of $\mathcal{F}^-$ around a point is equal to the distance of the point to the nearest $\mathbb{CPS}$ of $\mathcal{F}^{-1}$.

Since $\mathcal{F}$ is known, all derivatives of $\mathcal{F}$ can be known, and conversely all inverse derivatives of Υ^{-1} can known over non-singular partititions of $\mathcal{F}$. One has the option of construction of higher order Taylor approximations of Υ^{-1} but in multidimensions, the higher order curvature terms quickly become large matrices and a high accuracy approximation can be defined with multiple first order expansions if the Voronoi tesselations of $\aleph$ are small enough.

Let ${}^r\mathbf{x}_i^s$ be a point of the $\mathcal{F}^{-1}$ pre-image and let ${}^r_{\aleph_a}\mathbf{x}_i^s$ index by a all of the elements of the tesselation. Let ${}^r_{\aleph_c}\mathbf{x}_i^s$ be that element of the tesselation nearest to ${}^r\mathbf{x}_i^s$, i.e. $||{}^r\mathbf{x}_i^s - {}^r_{\aleph_c}\mathbf{x}_i^s|| < ||{}^r\mathbf{x}_i^s - {}^r_{\aleph_a}\mathbf{x}_i^s|| \forall a \in {}^r_{\aleph}\mathbf{x}_i^s$ and similarly for ${}_{\aleph_c}\zeta_{--}$ using a Riemannian metric, i.e. angular difference. Assuming that the optimal factorisation s has been derived in the canonical representation r, in order to reduce possible confusion with second and third order subscripting and superscripting of solution element prefixes and suffixes, let $\mathbf{w}_c = ({}_{\aleph_c}{}^r\mathbf{x}_i^s, {}_{\aleph_c}\zeta_{--})$ and $\mathbf{w}_*$ be an arbitrary point in the Voronoi cell, a dual hyper-cube tesselation when $\aleph$ is orthogonal, of $\aleph$ around $\mathbf{w}_c$. A first order multi-dimensional Taylor approximation of Υ^{-1} around $\mathbf{w}_c$ is defined by $\mathcal{F}^{-1}(\mathbf{w}_*) = \Upsilon^{-1}(\mathbf{w}_c) + \mathbf{J}^{\Diamond^{-1}}(\mathbf{w}_* - \mathbf{w}_c) + E_1(\mathbf{w}_*)$ where $E_1(\mathbf{w}_*)$ is the approximation error at $\mathbf{w}_*$ expressed in integral form as

$$E_1(\mathbf{w}_*) = \int_{\mathbf{w}_c}^{\mathbf{w}_*} \mathbf{H}^{\Diamond^{-1}}|_{\mathbf{t}}(\mathbf{w}_* - \mathbf{w}_c)d\mathbf{t} \tag{8.6}$$

where $\mathbf{H}^{\Diamond^{-1}}|_{\mathbf{t}}$ is the inverse augmented Hessian of $\mathcal{F}$ evaluated at $\mathbf{t}$ and $\mathbf{t}$ are dummy variables. (8.6) can be evaluated accurately since $\mathbf{H}^{\Diamond^{-1}}|_{\mathbf{t}}$ can be evaluated over $\mathbf{t}$ but it is outside the scope of this work to integrate $\mathbf{H}^{\Diamond^{-1}}|_{\mathbf{t}}$ analytically for large manipulators defined by $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathsf{SE}(3)$. Alternatively, by the second mean integral theorem, $E_1(\mathbf{w}_*)$ may also be expressed in Lagrange form as $E_1(\mathbf{w}_*) = \frac{1}{2}(\mathbf{w}_* - \mathbf{w}_c)^T\mathbf{H}^{\Diamond^{-1}}|_{\eta}(\mathbf{w}_* - \mathbf{w}_c)$ where η is is an (unknown) point between $\mathbf{w}_c$ and $\mathbf{w}_*$. Since η is unknown, the Lagrange expression of approximation error $E_1(\mathbf{w}_*)$ can not be exactly evaluated. However, error order can be estimated. In Landau's o-notation

$$E_1(\mathbf{w}_*) = o(||\mathbf{w}_c - \mathbf{w}_*||) \text{ as } ||\mathbf{w}_c - \mathbf{w}_*|| \rightarrow \mathbf{w}_c \iff \lim_{\mathbf{w}_* \rightarrow \mathbf{w}_c} \frac{E_1(\mathbf{w}_*)}{||\mathbf{w}_c - \mathbf{w}_*||} = 0. \tag{8.7}$$

See (Apostol, 1967) pg. 286 for proof. This means that $E_1(\mathbf{w}_*)$, i.e. error of $\mathcal{F}^{-1}(\mathbf{w}_*)$ approximation, is of smaller order than $(||\mathbf{w}_c - \mathbf{w}_*||)$. For $\mathbf{w}_*$ near to $\mathbf{w}_c$, $E_1(\mathbf{w}_*)$ is small compared to $||\mathbf{w}_c - \mathbf{w}_*||$. The corollary of this is that aproximation error $E_1(\mathbf{w}_*)$ scales linearly in $||\mathbf{w}_c - \mathbf{w}_*||$. Thus in order to reduce the bound of maximum $E_1(\mathbf{w}_*)$ to $o||\mathbf{w}_i - \mathbf{w}_*||$ between Voronoi edges of $\aleph$ tessellations, i.e. at switch between first order Taylor approximations around each $\mathbf{w}_i$ of $\aleph$, $\aleph$ density must be increased, d reduced where $d = ||\mathbf{w}_i - \mathbf{w}_j||$ for all regular tessellation neighbours i, j.

Let $L(\mathbf{w}_*)$ be a multi-dimensional linear interpolation[12] over the smallest tesselation subset bounding hypercube ${}^r_{\aleph_\square}\mathbf{x}_i^s$ of $\mathbf{w}_*$. This work proposes an method to reduce $E_1(\mathbf{w}_*)$ by a combination[13] of Taylor expansion approximations around $\mathbf{w}_i$ with $L(\mathbf{w}_*)$ over ${}^r_{\aleph_\square}\mathbf{x}_i^s$ such that

$$\mathcal{F}^{-1}(\mathbf{w}_*) = \frac{1}{2}L(\mathbf{w}_*) + \Upsilon^{-1}(\mathbf{w}_c) + \mathbf{J}^{\Diamond^{-1}}(\mathbf{w}_* - \mathbf{w}_c). \tag{8.8}$$

[12] Expression of a multi-dimensional linear interpolation in terms of coordinates is cumbersome and difficult to generalise above three dimensions. To avoid enumeration of various coordinate system specific expressions for the hyperplane, L is defined existentially. See (Press et al., 1992), pg. 123, for definition of bilinear interpolation over an $\mathbb{R}^2$ orthogonal lattice.

[13] The combination is not a nonparametric approximation since all the parameters of the linear interpolant and the Taylor approximation have an analytical interpretation, for instance in terms of covariant and contravariant tensor components.

(8.8) is generally a good approximation since between $\mathbf{w}_c$ and its Voronoi edges, Υ^{-1} is bounded by the Taylor approximation in (8.6) and Υ^{-1} is also bounded by the linear L interpolation to the neighbours. Υ^{-1} lies between the two hyper surfaces at least up to the Voronoi edge if there are no local maxima and local minima combinations of $\mathcal{F}$ over the domain of interpolation.

$\mathcal{F}^{-1}$ of (8.8) has at least half the error on average of either the hyper-plane interpolant or the stand-alone multi-dimensional Taylor approximation and the error bounds can be evaluated exactly without integration of Hessian and higher order approximation terms. For a tesselation interval small enough, there are no local maxima and local minima combinations of $\mathcal{F}$ over the Voronoi cell pre-image of (8.8) andE_{1l} of (8.8) is at most

$$E_{1l}^{\max} < \pm\frac{1}{2}||(\Upsilon^{-1}(\mathbf{w}_c) + \mathbf{J}^{\Diamond^{-1}}(\mathbf{w}_* - \mathbf{w}_c)) - L(\mathbf{w}_*)|| \tag{8.9}$$

and is generally of order $o(\mathbf{w}_* - \mathbf{w}_c)$. The maximum error of (8.8) approximation is exactly half the error of the linear interpolant minus half the error of the Taylor approximation and one average (8.8) is a good approximation of Υ^{-1}.

For a large tesselation interval, if $\mathcal{F}$ turns through a local maximum and a local minimum (relative to the interpolation interval), then $\mathcal{F}$ is not necessarily bounded between the linear interpolation hyper plane and the Taylor expansion hyperplanes over a given Voronoi cell around $\aleph$ expansion points.

The approximation of (8.8) requires special treatment at $\mathbb{CPS}$ where Υ switches from contractable to non-contractable disjoint solution homotopy. At $\mathbb{CPS}$, $\mathbf{J}^{\Diamond}$ looses rank and hence cannot be inverted. Furthermore close to $\mathbb{CPS}$, $\mathbf{J}^{\Diamond}$ ill-conditioning leads to increasing numerical errors in determination of $\mathbf{J}^{\Diamond^{-1}}$. In order to define $\mathcal{F}^{-1}$ over $\mathbb{CPS}$ seperating $\mathfrak{W}_w$, the $\mathbf{J}^{\Diamond^{-1}}$ term of (8.8) can be neglected[14]. In general for factorisations of the solution group of the pre-image of Υ down to a solution element pre-image homotopic to $\mathbb{R}^+$, the loss of *dof* will not occur along the direction of the solution element pre-image iteself, hence interpolation is well defined over the $\mathbb{CPS}$. The loss of *dof* occurs at the tangent plane[15] to the $\mathbb{CPS}$ for workspace interior $\mathbb{CPS}$. In general a rectilinear trajectory from the $\mathbb{CPS}$ to a point on this surface is not defined. This is a trajectory inversion constraint which is independent of the evaluation of $\mathcal{F}^{-1}$.

$\mathcal{F}^{-1}$ is continuous but not smooth at $\aleph$ induced Voronoi tessellation edges. A linear combination of two non-smooth functions is a non smooth function. The linear interpolants are non-smooth over the tesselation elements. The piecewise Taylor approximation is nonsmooth at the Voronoi tesselations. The combined interpolation and Taylor approximation is non smooth at the Voronoi tesselations. To make the approximation smooth, a smooth at least second order interpolation is required[16] between both interpolation elements and a neighbourhood of the interpolation elements of size of the order of the smoothing interpolation. Similarl arguments hold for interpolation between the Jacobians of the Taylor expansions. Such interpolations are available 'off the applied mathematical shelf' for regular lattices. There are at least two different means to extend bilinear interpolation to incorporate higher-order curvature information:

1. use of higher-order curvature estimates to obtain increased accuracy without smoothness,

2. use of higher-order curvature estimates to enforce smoothness of some of the derivatives as the interpolant crosses grid-square boundaries.

[14] Or perhaps replaced by a pseudoinverse of the augmented Jacobian inverse.

[15] Thus the loss of kinematic freedom of the manipulator is defined by a fibre bundle where the $\mathbb{CPS}$ is the base space and the tangent surfaces to the $\mathbb{CPS}$ form a bundle where inverse kinematic trajectory control is lost.

[16] The use of a Gaussian partition of unity of the Taylor elements, e.g. (5.31), is not a good idea since the error terms cannot be derived and inclusion of distant curvature information in a local expansion approximation results in solution evaluation ineffeciency.

The combination of the linear interpolant with the most local Taylor approximation is an example of the first approach. The more standard way to incorporate curvature information, without the simplicity of evaluation of the residual approximation error maxima, is described in (Press et al., 1992), p.g. 124. This is based on polynomial interpolated approximation of higher order curvature information. In the notation of (Press et al., 1992), if interpolation is defined on (x_1, x_2), to incorporate higher order curvature in interpolation, the interpolation problem is decomposed into a succession of one dimensional polynomial interpolations. If a $m-1$ order interpolation is required in the x_1 direction and a $n-1$ order interpolation is required in the x_2 direction, the $m \times n$ sub-block of the tabulated function matrix that contains (x_1, x_2) is located. m one dimensional polynomial f_m order $m-1$ interpolations are done in the x_2 direction to get function values at $y_m^1 - f_m(x_1)$ at x_2. Finally a $n-1$ order polynomial interpolation is performed on the points y_m^1 for all m and the resulting function evaluated at x_1. This type of sequential polynomial interpolation is not smooth across cell boundaries. To enforce interpolation smoothness, bicubic interpolation is required.

Bicubic interpolation requires the user to specify at each grid point the function values, Jacobian, and Hessian. Then an interpolation function that is cubic in the scaled coordinates of the interpolation coeffecients of the contribution of the grid points to the interpolation, can be found with the following properties:

1. the values of the function and the specified derivatives are reproduced exactly on the grid points,
2. the values of the function and the specified derivatives change continuously as the interpolating point crosses from one grid square to another.

An implementation of bicubic interpolation relies on certain coeffecients which have been historically determined. For an exact description of this algorithm, refer to the code provided in (Press et al., 1992), pg. 126 -127. A bicubic interpolation would generate a superior approximation to the proposed combination of hyper-plane interpolation with multi-dimensional multiple Taylor expansions approximation if approximation smoothness is a primary functional requirement. The generalisation of both approaches to higher than two dimensional grids is analogous but the numerical housekeeping on indexes becomes cumbersome with higher order matrices embedded in two dimensional matrices.

8.2 Approximation of the Canonical Solution Groups of Planar 3-R

The kinematic mapping of planar 2-**R** ($\mathcal{F} : \mathsf{T}^2 \rightarrow \mathsf{R}^2$) and planar 3-**R** ($\mathcal{F} : \mathsf{T}^3 \rightarrow \mathsf{R}^2$) are defined and inverted in closed form in Appendix D and Appendix E respectively. In this section, a single canonical Lie factor cyclic solution branch factor is derived for both of these manipulators and a local linear approximation of the solution element of planar 3-**R** is defined.

8.2.1 Geometric Factorisation of the Canonical Solution Groups

The pre-image of Υ^{-1} of planar 2-**R** and planar 3-**R** is a subset of $\mathbb{R}^{+2}$. $\mathbb{R}^{+2}$ can be factored by $\mathbb{R}^+ \times \mathsf{S}^1$ by Lemma 8.1. $\mathbb{R}^+ \times \mathsf{S}^1$ define a polar coordinatisation of $\mathbb{R}^2$. The polar coordinatisation of a point P in a Cartesian coordinate system $\{X_0, Y_0\}$ is represented by $(R, \beth)$ where R is the length of the line segment joining P to the origin, and $\beth$ is the angle between the line segment and the X_0 axis. T^2 and T^3 can also be factored by S^1 by Definition 7.31. If $(R, \beth_j)$ correspond to (θ_1^j, θ_2^j) and $(R, \beth_i)$ correspond to (θ_1^i, θ_2^i), geometric analysis of planar 2-**R** in Figure 2.3 reveals that a transformation

$\theta_1^i = \theta_1^j + \Delta\theta_1$ of the shoulder joint induces a transformation of the position $\rceil_j$ to $\rceil_i = \rceil_j + \Delta\rceil$ where $\Delta\rceil = \Delta\theta_1$. This defines a solution group $\mathsf{G}_1 = ({}^{r*}\Psi_1^{s*}, {}^{r*}\Phi_1^{s*})$ such that

$$ {}^{r*}\Phi_1^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \rceil_i = \rceil_j + \Delta\rceil, \tag{8.10} $$

$$ {}^{r*}\Psi_1^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_1^i = \theta_1^j + \Delta\rceil \tag{8.11} $$

and

$$ \theta_j^{s*} =_j^{r*} \Psi_1^{s*^{-1}}(F^{-1}({}_j^{r*}\Phi_1^{s*}(\mathbf{x}_j^{s*}), \zeta)). \tag{8.12} $$

${}^{r*}\Phi_1^{s*}$ is a canonical solution group operator that transforms ${}^{r*}\mathbf{x}_j^s \in \mathbb{R}^{+2}$ to ${}^{r*}\mathbf{x}_i^s \in \mathbb{R}^{+2}$ and ${}^{r*}\Psi_1^{s*}$ defines the inverse transformation on the solution fibre bundle. Comparison of Figure 2.3 and Figure 8.1 reveals that the same G_1 also holds for planar 3-$\mathbf{R}$.

${}^{r*}\Phi_1^{s*}$ involves a single addition. This indicates that it is the canonical solution group operator representation. This is confirmed by comparison with the representation of this solution group orbit in an orthonormal Cartesian coordinatisation of $\mathbb{R}^2$. For transformation of (x_j, y_j) corresponding to $(R, \rceil_j)$ to (x_i, y_i) corresponding to $(R, \rceil_i)$, the representation of ${}^{r*}\Phi_1^{s*}$ of (8.10) in an orthonormal representation of Cartesian space is

$$ {}^{r}\Phi_1^{s*} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 : \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \cos(\Delta\rceil) & -\sin(\Delta\rceil) \\ \sin(\Delta\rceil) & \cos(\Delta\rceil) \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}. \tag{8.13} $$

(8.10) is simpler, hence more canonical than, (8.13). Geometric analysis of planar 2-$\mathbf{R}$ in Figure 2.3 and examination of the analytical solution in Appendix D also reveals the solution branch factor for $\mathsf{C}_2 = ({}^{r*}\Phi_2^{s*} = \mathbf{I}, {}^{r*}\Psi_2^{s*})$. For $(R, \rceil_j)$,

$$ {}^{r*}\Psi_2^{s*} : \theta_2^{j^2} = -\theta_2^{j^1}, \tag{8.14} $$

$$ \theta_1^{j^2} = \rceil - (\theta_1^{j^1} - \rceil) = 2\rceil - \theta_1^{j^1}. \tag{8.15} $$

(8.15) is simpler, hence more canonical, than (D.7). The canonical solution branch factor for planar 3-$\mathbf{R}$ with the $\zeta = \theta_1$ parameterisation proposed in Appendix E is different for θ_2 and the same for θ_1

$$ {}^{r*}\Psi_2^{s*} : \theta_2^{j^2} = \theta_2^{j^1} - \pi \tag{8.16} $$

$$ \theta_1^{j^2} = \rceil - (\theta_1^{j^1} - \rceil) = 2\rceil - \theta_1^{j^1}. \tag{8.17} $$

See (E.24) and (E.25). The solution element of planar 2-$\mathbf{R}$ under a $\mathsf{G}_1, \mathsf{C}_2$ factorisation is (R, θ_1, θ_2), i.e. a radius R in $\mathfrak{W}$ with an associated single solution point (θ_1, θ_2) for each point of R. Thus the solution fibre under the $\mathsf{G}_1, \mathsf{C}_2$ factorisation of $\mathcal{F}$ of planar 2-$\mathbf{R}$ is the most trivial fibre - a single point. The solution fibre under the $\mathsf{G}_1, \mathsf{C}_2$ factorisation of $\mathcal{F}$ of planar 3-$\mathbf{R}$ (R, S), i.e. a radius R with an associated solution fibre homotopic to S over each point of R.

8.2.2 The Homotopy of the Trivial Fibre Bundle Solution Element

8.2 Definition (The Solution Fibre of Planar 3-R)
Let $\mathfrak{W}_w$ be maximal $\mathbb{CPS}$-free regions of $\mathfrak{W} \subset \mathbb{R}^2$. Let $\theta_i \subset \mathsf{T}^3$ be a connected region such that $\Upsilon(\theta_i) = \mathfrak{W}_i$. $\forall \mathbf{x}_i \in \mathfrak{W}_i, \Upsilon^{-1}(\mathbf{x}_i, \zeta)$ is diffeomorphic to S^1 (DeMers, 1993).

The global fibre bundle solution on base space $\mathfrak{W}$ is a union of trivial fibre bundle solutions on $\mathfrak{W}_w, w = 1, \ldots, 4$. Each $\mathfrak{W}_w$ is a maximally connected subset of $\mathfrak{W}_w$ without $\mathbb{CPS}$. $\mathfrak{W}_w$ are partitioned by $\mathbb{CPS}$. $\mathbf{J}$ of planar 3-$\mathbf{R}$ looses rank at $\mathbb{CPS}$. The $\mathbb{CPS}$ have annular structure. At a point in

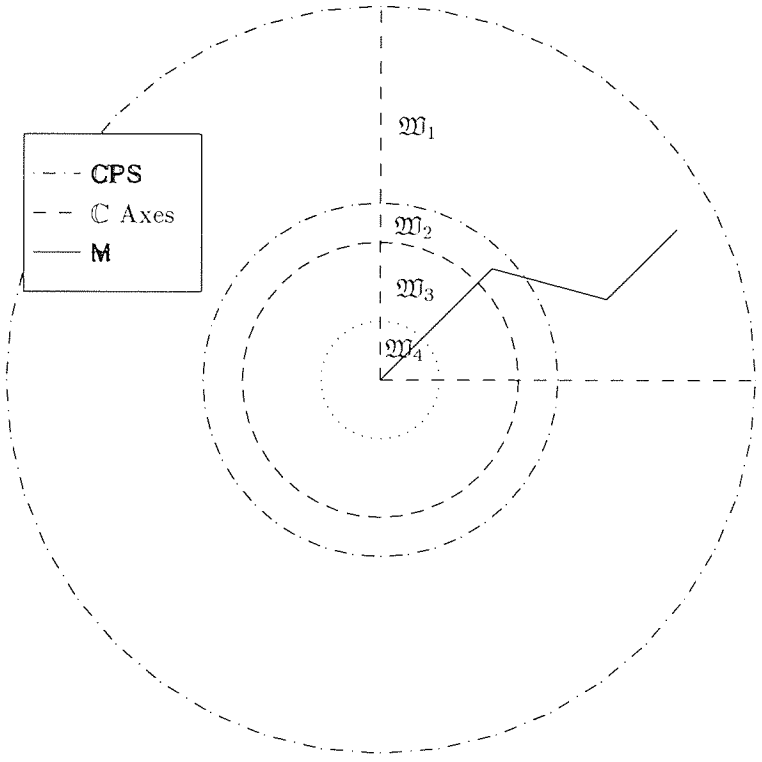

Figure 8.1: The Homotopy Partitions of Cartesian Base Space of a Planar 3-**R**.

the $\mathbb{CPS}$, planar 3-**R** cannot move along the tangent to the $\mathbb{CPS}$ at that point. Planar 3-**R** has two possible homotopy classes for solution fibres - contractable or disjoint non-contractable. The $\mathbb{CPS}$ is where fibre homotopy structure changes from contractable to non-contractable disjoint. The topological structure of $\Upsilon^{-1}(\mathbf{x}_i, \zeta)$ can only change at $\mathbb{CPS}$, hence $\Upsilon^{-1}(\mathbf{x}_i, \zeta), \mathbf{x}_i \in \mathfrak{W}_w$ is diffeomorphic to $\Upsilon^{-1}(\mathbf{x}_j, \zeta), \mathbf{x}_j \in \mathfrak{W}_w$.

The homotopy structure of $\mathcal{F}$ of planar 3-**R** over $\mathfrak{W}$ depends on the specific link lengths. Planar 3-**R** has 3 or 4 $\mathbb{CPS}$ depending on l_1, l_2, l_3. If the link lengths are such that the sum of any two are longer than the third, then the manipulator can reach its base, and the workspace is a disc of radius $l_1 + l_2 + l_3$. This subset of planar 3-**R** manipulators have $\mathfrak{W}_1, \mathfrak{W}_2, \mathfrak{W}_3, \mathfrak{W}_4$ annular regions separated by $\mathbb{CPS}$ such that

$$\mathfrak{W}_1 : l_1 + l_2 - l_3 < x^2 + y^2 < l_1 + l_2 + l_3, \tag{8.18}$$

$$\mathfrak{W}_2 : l_1 - l_2 + l_3 < x^2 + y^2 < l_1 + l_2 - l_3, \tag{8.19}$$

$$\mathfrak{W}_3 : l_1 - l_2 - l_3 < x^2 + y^2 < l_1 - l_2 + l_3, \tag{8.20}$$

$$\mathfrak{W}_4 : 0 < x^2 + y^2 < l_1 - l_2 - l_3. \tag{8.21}$$

See Figure 8.1. Table 8.2.2 describes the homotopy classes for planar 3-**R** that is analysed in Appendix E and which also appears in Figure 8.1.

An algebraic solution for a $\zeta = \theta_3$ parameterisation of the solution fibre bundle of planar 3-**R** is derived in Appendix E. With a $\zeta = \theta_3$ coordinatisation of the fibre solution, two solution values are possible for disjoint fibres and also for contractable fibres depending on the value of θ_2. In the case of contractable fibres in $\mathfrak{W}_1, \mathfrak{W}_3$, for each θ_3, there are two solutions θ_2^1, θ_1^1 and θ_2^2, θ_1^2. Let these solutions

be denoted elbow-up and elbow-down. The fibres in $\mathfrak{W}_1, \mathfrak{W}_3$ have two solution switch-over[17] points. The switch-over points occur when $\theta_2 = 0$. At switch-over points, elbow up and elbow down solutions are the same. Hence $\theta_2 = 0$ partitions the fibre of $\mathfrak{W}_1$ into an elbow up functions and an elbow down solution functions. Elbow up and elbow down solution functions do not correspond to solution branches since the fibre is not disjoint. Up and down solutions are not elements of two distinct solution branches by Definition 7.14. Since the fibre is contractable, each of these solutions may be converted to the other one by traversal of the fibre. The non disjoint fibres have two switch-over points where it is possible to switch from arm-up solution to arm-down solution. At the $\mathbb{CP}\mathbb{S}$ the two switch-over points collapse into a single switch-over point and there are no switch-over point in the disjoint fibre bundles after the solution fibre bifurcates. See Figure 8.3 and Figure 8.5. At the $\mathbb{CP}\mathbb{S}$ partition of $\mathfrak{W}_1$ and $\mathfrak{W}_2$, the two fibre switch-over points contract to a single point corresponding to $\theta_2 = 0, \theta_1 = \pi$. There is only a single location where elbow up functions can change to elbow down functions.

Solution fibres are plotted in Figure 8.2 and Figure 8.4 using the exact inverse solution derived in Appendix E. Figure 8.2 shows the solution fibre at a $\mathbb{CP}\mathbb{S}$ at $x = l_1 + l_2 - l_3, y = 0$ for a manipulator with link parameters the same as those used in (DeMers, 1993) $l_1 = 0.4m, l_2 = 0.3m, l_3 = 0.25m$. This manipulator has 4 $\mathbb{CP}\mathbb{S}$. $\mathbb{CP}\mathbb{S}$ are independent of θ_1. See Figure 8.1. The fibre in Figure 8.2 has only one switch-over point and is about to undergo bifurcation into a disjoint manifold. Figure 8.3 plots a discretisation of fibres solutions in base space at the $\mathbb{CP}\mathbb{S}$ between $\mathfrak{W}_1$ and $\mathfrak{W}_2$. If the fibre does not look like canonical S^1, this is due to the wraparound of the fibre through the axes that have been chosen. The elbow up/down solution changes at switch-over points.

Base Sub-space	Fibre Structure	Homotopy Class
$\mathfrak{W}_1$	not disjoint	contractable.
$\mathfrak{W}_2$	disjoint in θ_2	non-contractable along θ_3
$\mathfrak{W}_3$	not disjoint	contractable.
$\mathfrak{W}_r$	disjoint in θ_3	non-contractable in θ_1

Table 8.1: The Homotopy Structure of Trivial Fibre Bundles of Planar 3-**R**. See also (DeMers, 1993).

[17] Solution switch-over points correspond to fibre end points identified in §8.1.3.1.

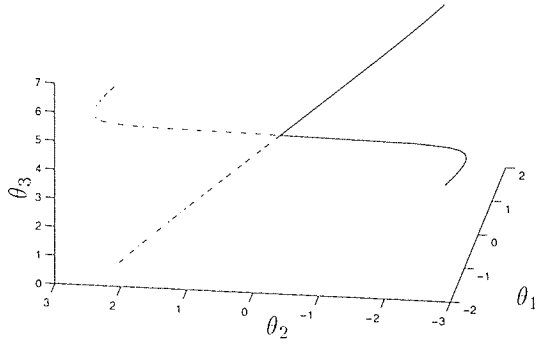

Figure 8.2: The solution fibre of Planar 3-**R** at $x = l_1 + l_2 - l_3; y = 0 \in \mathbb{CP}^1$. The fibre embedding space T^3 is unfolded by cutting $\theta_1, \theta_2, \theta_3$. Each face of the cube wraps around to the corresponding opposite face. There is only one switch-over point at the $\mathbb{CP}^1$.

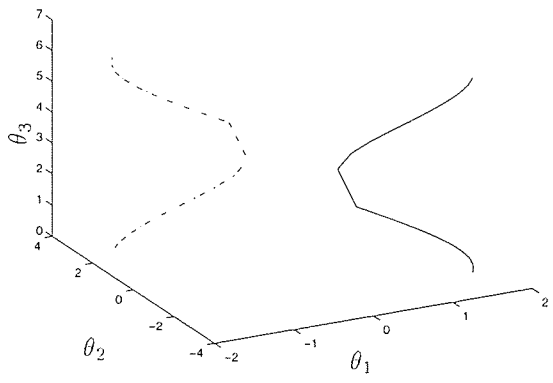

Figure 8.4: The solution fibre at $x = l_1 + l_2 - l_3 - 0.1; y = 0$. The fibre embedding space T^3 is unfolded by cutting $\theta_1, \theta_2, \theta_3$. Each face of the cube wraps around to the corresponding opposite face. The solution fibre is disjoint.

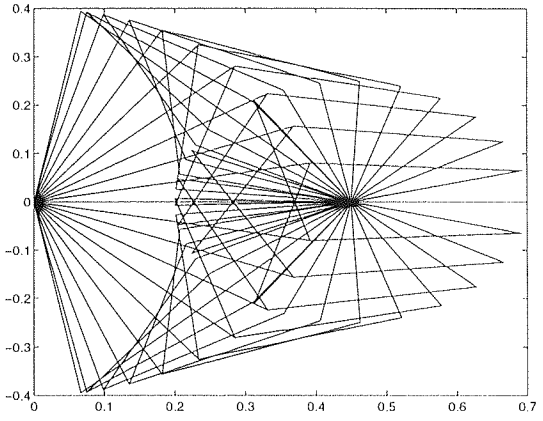

Figure 8.3: A Plot of the forward kinematics of Planar 3-**R** for some fibre solutions at $x = l_1 + l_2 - l_3; y = 0$.

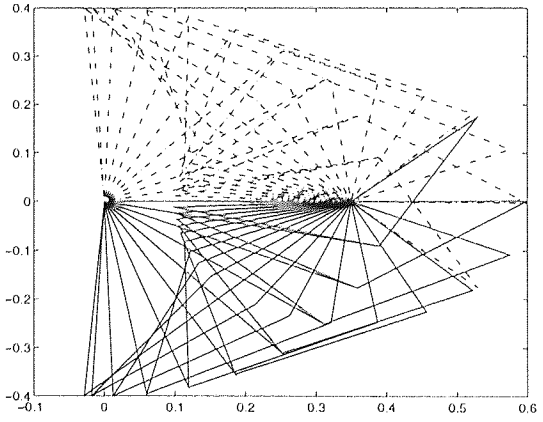

Figure 8.5: A plot of the forward kinematics of planar 3-**R** for some fibre solutions t $x = l_1 + l_2 - l_3 - 0.1; y = 0$.

	Error on θ_1			Error on θ_2		
Error	Linear(r)	Taylor(r)	Combined(r)	Linear(r)	Taylor(r)	Combined(r)
max	0.0043	0.0129	0.0013	0.1743	0.1319	0.0658
mean	-0.0007	0.0005	-0.0001	-0.0018	0.0045	0.0013
std	0.0065	0.0033	0.0017	0.0531	0.0265	0.0162

Table 8.2: The Radian Error Statistics on the Test Data for the Linear Interpolant, the Taylor Approximation, and the combined Linear Interpolant and Taylor Approximation.

In $\mathfrak{W}_2$ there are no elbow up function and elbow down function switch-over points. Elbow up and elbow down functions in $\mathfrak{W}_2$ correspond to the two disjoint pieces of $\mathfrak{F}$. Figure 8.4 shows the disjoint fibre solution for $x = l_1 + l_2 - l_3 - 0.1, y =$ inside $\mathfrak{W}_2$ of the same manipulator. Figure 8.5 plots a discretisation of fibre solution in base space at this point. Different solution branches are indicated by a solid line and a dashed line. See also (DeMers, 1993) for additional visualisation plots of the fibre homotopies of planar 3-**R**.

8.2.3 A Local Linear Approximation of the Solution Element

Since the preimage of Υ^{-1} is homotoptic to $\mathbb{R}^+ \times \mathrm{S}$, i.e. two dimensional, the local linear approximation proposed in §8.1.3.2 is a combination of bilinear interpolation $L(\mathbf{w}_*)$ and a two dimensional Taylor approximation $\Upsilon^{-1}(\mathbf{w}_c) + \mathbf{J}^{\Diamond^{-1}}(\mathbf{w}_* - \mathbf{w}_c)$ within the Voronoi cell of the nearest $\aleph$ tesselation element. See (8.8). The bilinear interpolation of (Press et al., 1992), pg. 123, was implemented. $\mathbf{J}^{\Diamond}$ and $\mathbf{J}^{\Diamond^{-1}}$ are derived analytically in Appendix E. See (E.27). $\mathbf{J}^{\Diamond^{-1}}$ can by analytically evaluated at any location on the solution element in order to construct Γ.

In order to test the approximation, a set of test points was generated for $x = (0.38, 0.41, 0.42)$ and θ_3 sampled from 1.0472 to 5.8804 in increments of $\frac{\pi}{13}$. A local linear interpolant estimate of the test data, a Taylor approximation of the test data and a combined local linear interpolant and Taylor approximation of the test data were generated. See Figure 8.8 to Figure 8.13. Comparison of Figure 8.6 with Figure 8.8, Figure 8.10, and Figure 8.12 and a comparison of Figure 8.7 with Figure 8.9, Figure 8.11, and Figure 8.13 show that all three methods provide a reasonable approximation of Υ^{-1} over the low curvature sections. The accuracy of approximation diverges as curvature of Υ^{-1} increases. A comparison of Figure 8.14 with Figure 8.15 combined with a comparsion of Figure 8.16 with Figure 8.17 reveals that the combined local linear plus Taylor approximation has lower error over regions of high Υ^{-1} curvature. Figure 8.18 and Figure 8.19 show the errors of the local linear interpolation minus the combined local linear interpolant plus Taylor approximation. Histogram and scatter analyses of these error differences confirm that the combined local linear interpolant and Taylor approximation has lower error than the linear interpolation for all data points. See Figure 8.20 to Figure 8.25.

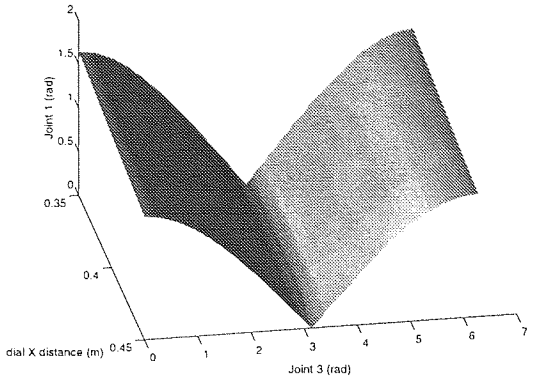

Figure 8.6: A Surface Plot of $\theta_1 = \Upsilon^{-1}(x, \theta_3)$ for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

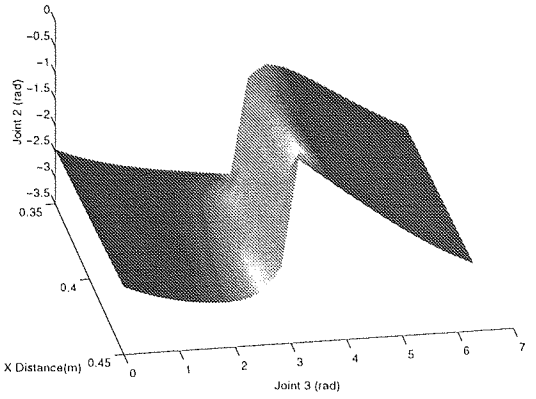

Figure 8.7: A Surface Plot of $\theta_2 = \Upsilon^{-1}(x, \theta_3)$ for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

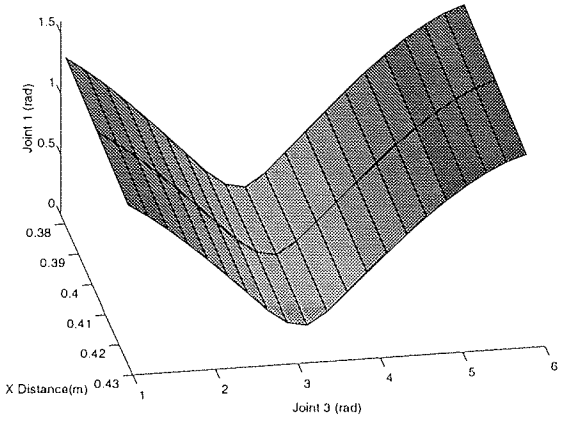

Figure 8.8: A Surface Interpolation Between the Linear Interpolation Values for the Test Data for θ_1 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

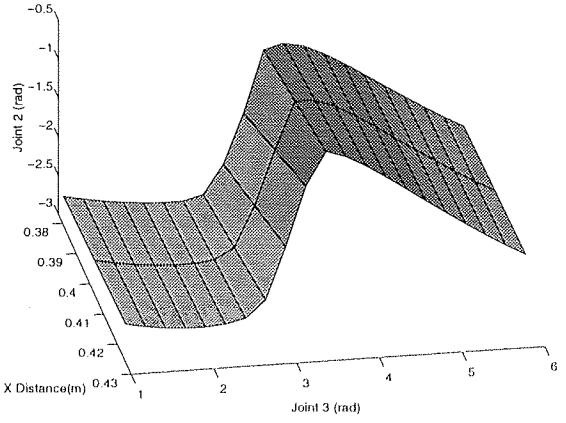

Figure 8.9: A Surface Interpolation Between the Linear Interpolation Values for the Test Data for θ_2 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

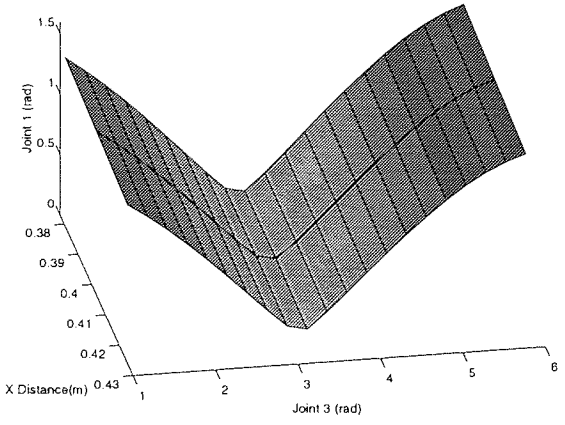

Figure 8.10: A Surface Interpolation Between the Taylor Approximation Values for the Test Data for θ_1 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

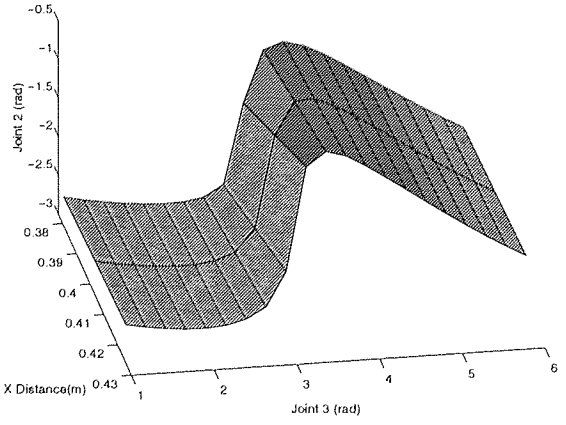

Figure 8.11: A Surface Interpolation Between the Taylor Approximation Values for the Test Data for θ_2 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

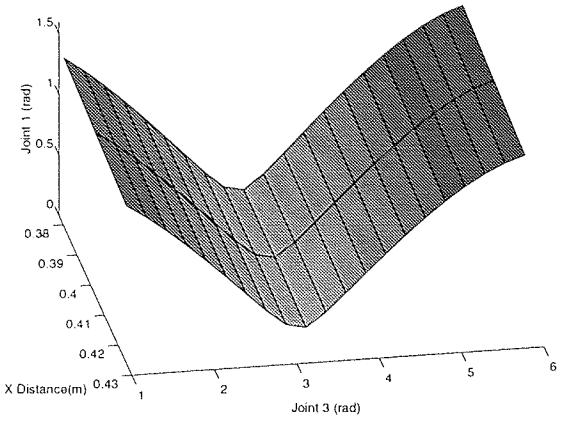

Figure 8.12: A Surface Interpolation Between the Combined Local Linear Interpolation plus Taylor Approximation Values for the Test Data for θ_1 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

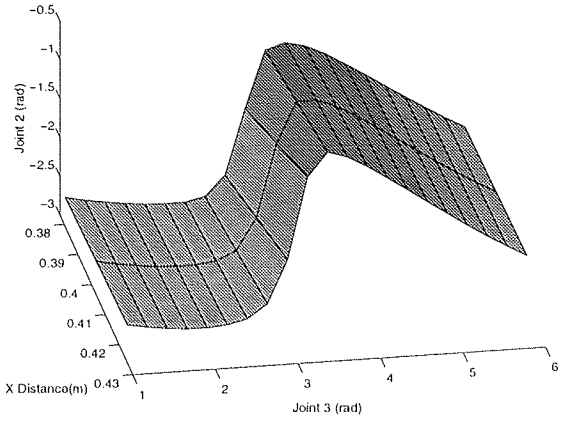

Figure 8.13: A Surface Interpolation Between the Combined Local Linear Interpolation plus Taylor Approximation Values for the Test Data for θ_2 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

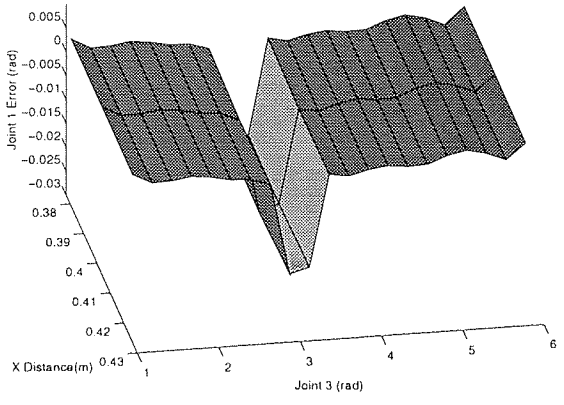

Figure 8.14: A Surface Interpolation of the Error of the Linear Interpolation Values for the Test Data for θ_1 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

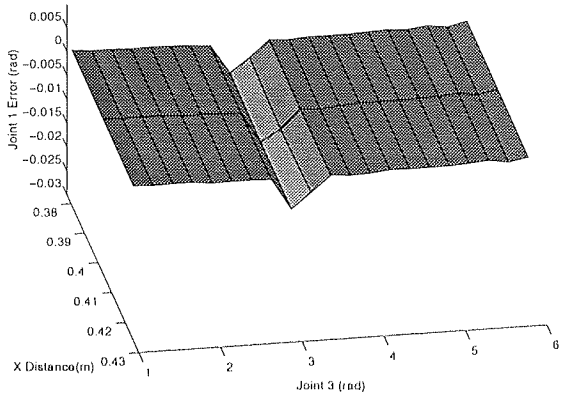

Figure 8.15: A Surface Interpolation of the Error of the Combined Linear Interpolant plus Taylor Approximation for the Test Data for θ_1 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

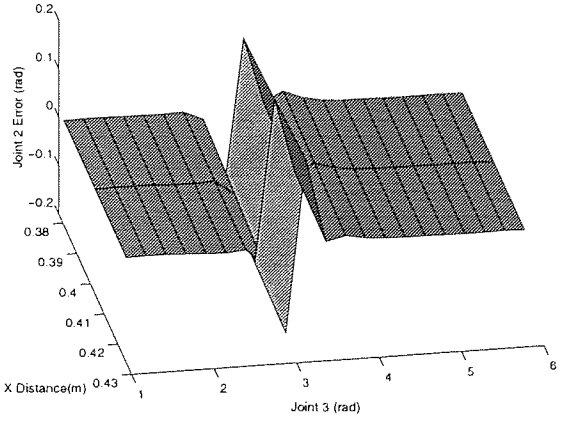

Figure 8.16: A Surface Interpolation of the Error of the Linear Interpolation Values for the Test Data for θ_2 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

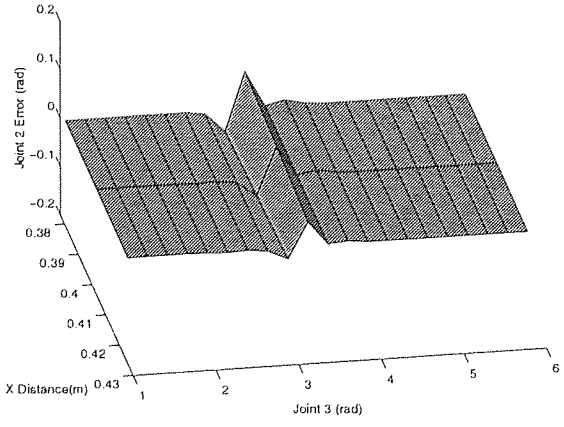

Figure 8.17: A Surface Interpolation of the Error of the Combined Linear Interpolant plus Taylor Approximation for the Test Data for θ_2 for the Elbow Up Solution Branch in $\mathfrak{W}_2$.

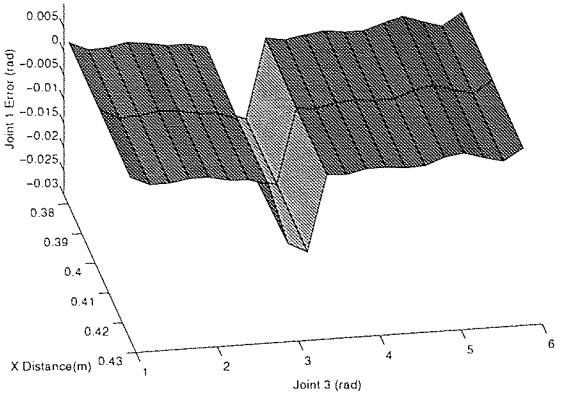

Figure 8.18: A Surface Interpolation of the Error of the Linear Interpolant Minus the Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_1.

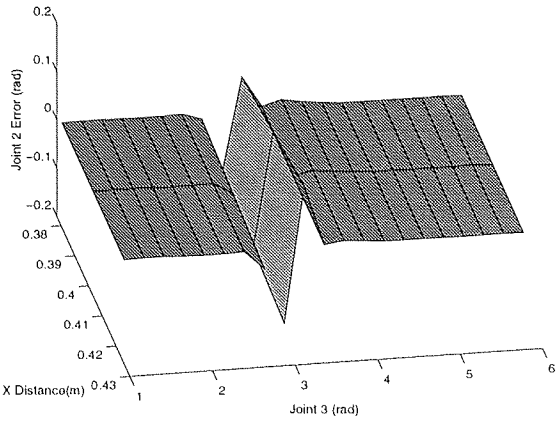

Figure 8.19: A Surface Interpolation of the Error of the Linear Interpolant Minus the Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_2

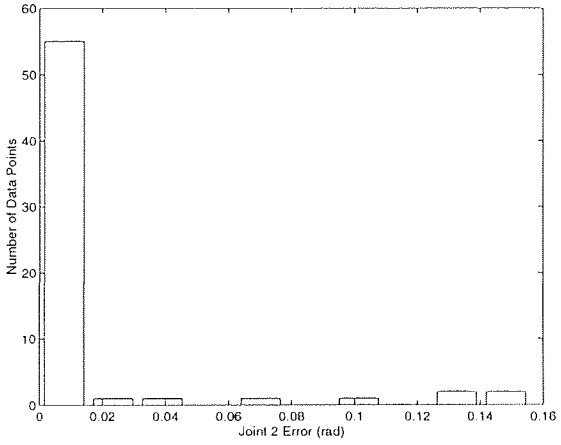

Figure 8.20: A Histogram of the Absolute Error of the Linear Interpolant minus the Absolute Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_2.

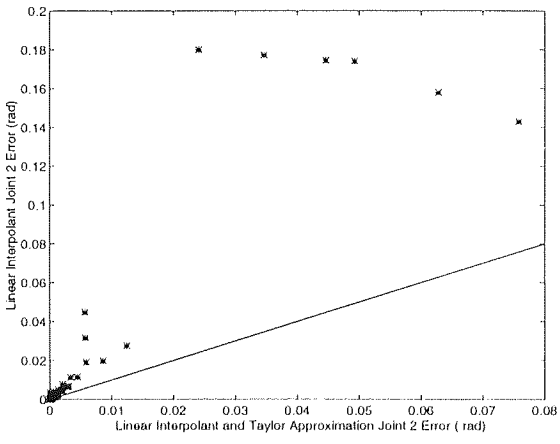

Figure 8.21: A Scatterplot of the Absolute Error of the Linear Interpolant against the Absolute Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_2.

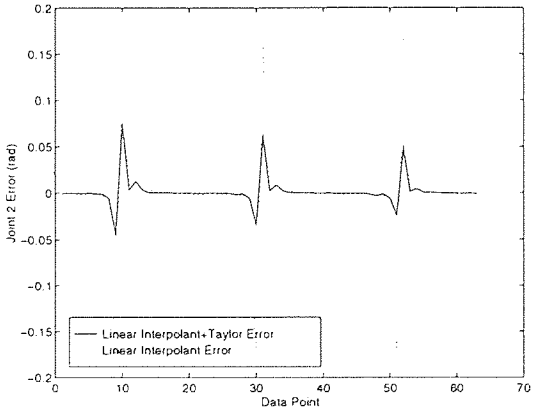

Figure 8.22: A Plot of the Error for θ_2 of the Linear interpolant and the Error of the Combined Linear Interpolant plus Taylor Approximation on each of the Test Data points.

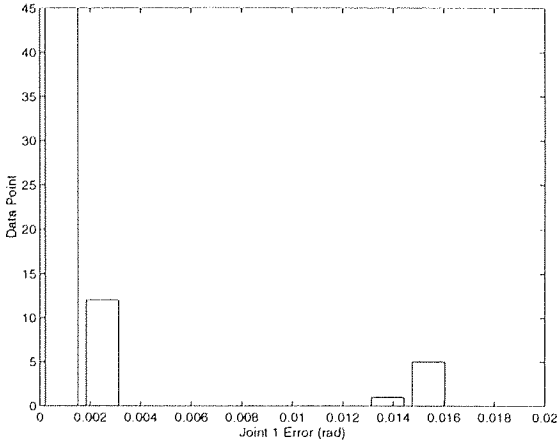

Figure 8.23: A Histogram of the Absolute Error of the Linear Interpolant minus the Absolute Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_1.

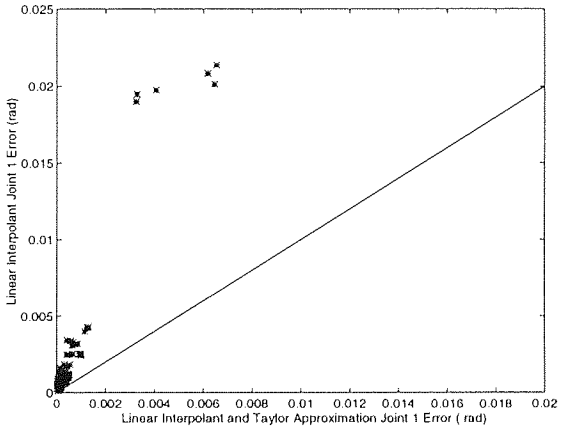

Figure 8.24: A Scatterplot of the Absolute Error of the Linear Interpolant against the Absolute Error of the Combined Linear Interpolant plus Taylor Approximation on the Test Data for θ_1.

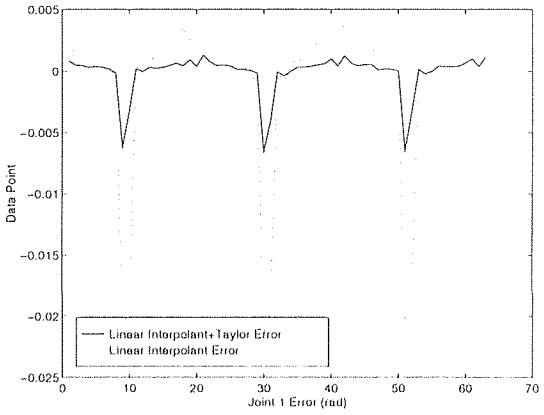

Figure 8.25: A Plot of the Error for θ_1 of the Linear interpolant and the Error of the Combined Linear Interpolant plus Taylor Approximation on each of the Test Data points.

Examination of Table 8.2 shows that the combined linear interpolation plus Taylor approximation has lower mean error, lower maximum error, and lower error standard deviation than either a local linear or a Taylor approximation for θ_1 and θ_2. On average over all coordinates, the combined approximation is twice as accurate as the linear interpolant and three times more accurate than the Taylor approximation. The maximum error of the combined local linear plus Taylor approximation is 0.0671. This is 2.65 times lower than the linear interpolant summed maximum error of 0.1783 and and 2.15 times lower than the Taylor approximation summed maximum error of 0.1448. This agrees well with the error analysis of (8.9).

Converting from radians to degrees, the combined linear interpolant and Taylor approximation has on average error less than one minute of a degree. Thus solution group action on this approximation will generate a highly accurate solution over the complete workspace $\mathfrak{W}$. Solution group factorisation of planar 3-**R** can be extended to higher dimensional dextrous manipulators. However, analytical evaluation of $\mathbf{J}^{\Diamond^{-1}}$ is not as simple as for planar 3-**R** to invert $\mathbf{J}^{\Diamond}$ of 7 *dof* dextrous manipulators.

8.3 Approximation of the Canonical Solution Groups of Seven *dof* Manipulators

Seven *dof* dextrous manipulators such as the Odetics/NASA, Robotics Research[18] K-1207 and Hollerbach arms (3**R**-**R**-3**R**) have better dynamic manipulability properties and stronger wrists than older spatial manipulator designs such as the PUMA 560. The kinematic design of a dextrous manipulator is a result of optimisation of multiple objective functions, for instance minimisation of singularity structures and wear and tear, maximisation of torque isotropy, repeatability and ease of parts maintenance. The final kinematic design of an advanced dextrous manipulator may or may not be closed form parameterisable and this can be an optimisation criterion for design of advanced manipulator geometries.

The anthropomorphic Hollerbach and RR-K-1207 manipulators have a shoulder roll θ_1 and pitch θ_2, elbow roll θ_3 and pitch θ_4, wrist roll θ_5 and pitch θ_6 and tool plate roll θ_7 joints. See Figure 8.26. The elbow roll joint and the lack of a yaw joint to orientate manipulator Cartesian yaw are the main structural kinematic differences between the Hollerbach and RR-K-1207 manipulators and spatial manipulators such as the PUMA 560. Since the Hollerbach and RR-K-1207 manipulators have no yaw joint, they cannot be arbitrarily Cartesian yaw oriented. A given position of these manipulators in a workspace can only have a single Cartesian yaw configuration. Thus the Cartesian space of these manipulators is in effect homotopic to $\mathbb{R}^{+3} \times \mathsf{SO}(2)$ with Cartesian yaw λ_j always constrained to be equal to θ_1. Thus the kinematic mapping of these manipulators is effectively homotopic to $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathsf{R}^3 \times \mathsf{SO}(2)$. Thus these manipulators have two effective degrees of redundancy to operate in $\mathsf{R}^3 \times \mathsf{SO}(2)$.

The Hollerbach manipulator DH parameters $a_4, a_5, a_6 = 0$, i.e it has zero offset from the central alignment axis. See Figure 8.27 and (Kreutz-Delgado et al., 1992). This simplifies kinematic analysis. If the value of any joint ζ_{θ_k} of the Hollerbach manipulator with no wrist offsets is fixed, the manipulator becomes a solvable spatial manipulator. See §2.5.1. For a given fixed value of ζ_{θ_k}, let the closed form solution of the DH table of the Hollerbach manipulator with ζ_{θ_k} fixed, DH$|\zeta_{\theta_k}$ be denoted $\mathcal{F}^{-1}_{\zeta_{\theta_k}}$. ζ_{θ_k} parameterises a manifold of closed form solutions $\mathcal{F}^{-1}_{\zeta_{\theta_k}}$.

The kinematic structure of a RR-K-1207 is more complicated that that of the Hollerbach arm.

[18] Robotics Research supply the RR-K-1207 for approximately 0.5m dollars per unit. The units are designed for critical space and nuclear applications such as hazardous waste disposal.

See Table 8.3 from (Kreutz-Delgado et al., 1992) and Figure 8.27. The RR-K-1207 DH parameters $a_4, a_5, a_6 \neq 0$, i.e. the arm has non-shoulder link offsets from the X, Z plane centred at $x = 0, z = 0, y = 0$ at the base of the manipulator.

The kinematics of a RR-K-1207 is closed form unsolvable.[19] Fixing a ζ_{θ_k} redundancy parameterisation joint of the RR-K-1207 converts into a general spatial manipulator with wrist offsets. This can be iteratively solved by the (Manocha and Canny, 1994) iterative eigendecomposition of the matrix polynomial representation of spatial manipulator kinematic equations. See §2.5.3.

link	joint	twist	length	offset
i	θ_i^o	α_{i-1}^o	a_{i-1}(cm)	d_i (cm)
1	θ_1	0	0	0
2	θ_2	-90^0	-12.319	0
3	θ_3	90^0	10.795	54.61
4	θ_4	-90^0	-7.938	0
5	θ_5	90^0	-7.938	54.61
6	θ_6	-90^0	-4.92	0
7	θ_7	90^0	4.92	0

Table 8.3: The Denavit Hartenberg Table of a Robotics Research K-1207.

(Kreutz-Delgado et al., 1992) propose a scalar parameterisation of the kinematic redundancy of the Hollerbach and the RR-K-1207 and a differential augmented Jacobian solution based on this scalar parameterisation. (Kreutz-Delgado et al., 1992) analyse the interaction of the $\mathbb{CPS}$ of these manipulators with the cyclicity properties of their proposed differential inversion of the scalar redundancy parameterisation augmented Jacobian solution. However it seems their solution method is underdetermined unless a 2 dimensional vector parameterisation of redundancy is used to augment the Jacobian since the RR-K-1207 have two degrees of redundancy since their forward kinematic mapping is effectivly $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathbb{R}^3 \times \mathsf{SO}(2)$. This work proposes a 2 dimensional vector direct parameterisation of the T^2 solution manifold of these manipulators.

[19] The Argand plane based exact solution of planar 3-$\mathbf{R}$ derived in Appendix E does not scale to inversion of $\mathsf{T}^7 \rightarrow \mathsf{SE}(3)$ mappings. The unsolvability of polynomials of order greater than four implies that such solutions do not exist unless dextrous manipulator equations can be combined, reduced and transformed to order less than five polynomials for insance in the case of the Hollerbach arm by fixing ζ_{θ_7}.

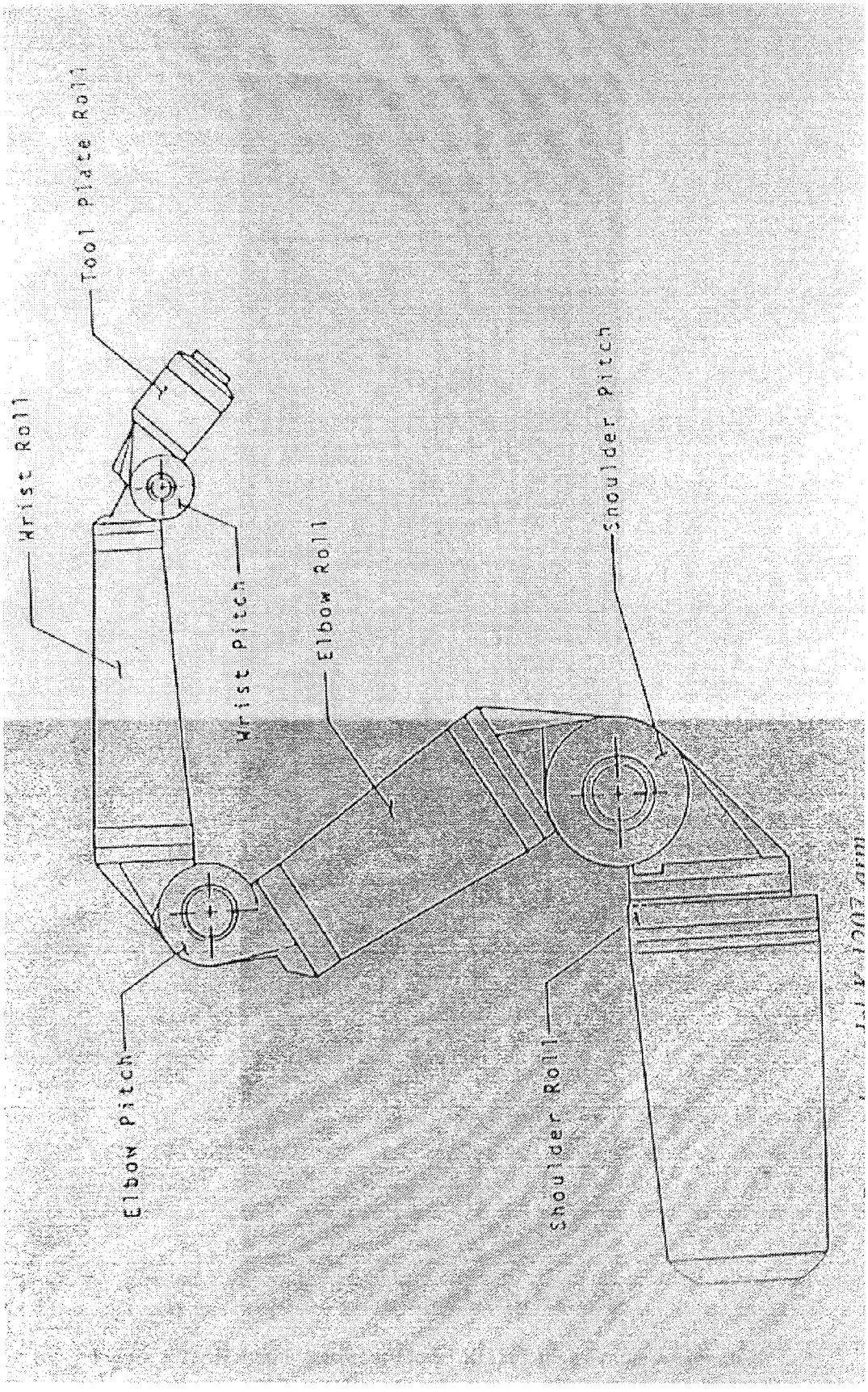

Figure 8.26: The Robotics Research Model K-1207 Arm.

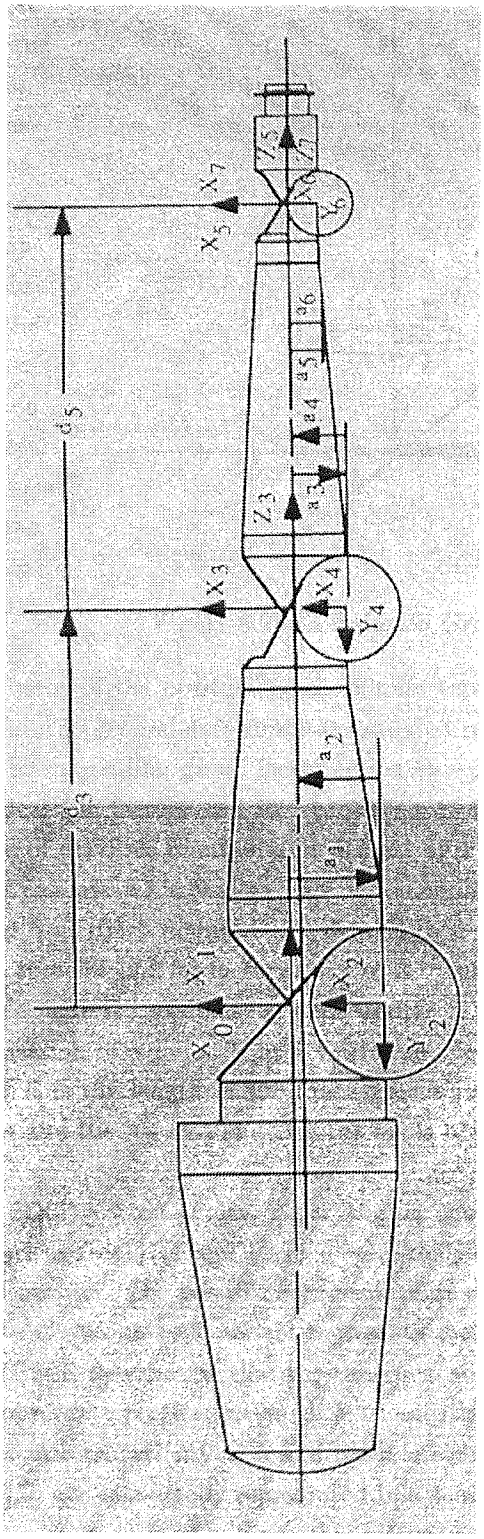

Figure 8.27: Robotics Research Model K-1297 Link Frame Assignment.

8.3.1 Geometric Factorisation of the Canonical Solution Groups

The canonical factorisation of the Planar 3-**R** solution group can be extended for the Hollerbach and RR-K1207 manipulators. The extension of the canonical factorisation requires the effects of the solution group actions on the orientation coordinates of the manipulator to be determined. A solution group operator action on the Cartesian position coordinates of these manipulators also acts on the orientation coordinates since orientation coordinates are coupled[20] to positioning coordinates by the kinematic structure of the manipulators. Positioning coordinates are not coupled to orientation coordinates. A solution group factor of the positioning coordinate representations of the $\mathbb{R}^{+3}$ factor group of SE(3) of the manipulator in general acts simultaneously on manipulator position and orientation coordinates for a given corresponding group factor action on a joint coordinate.

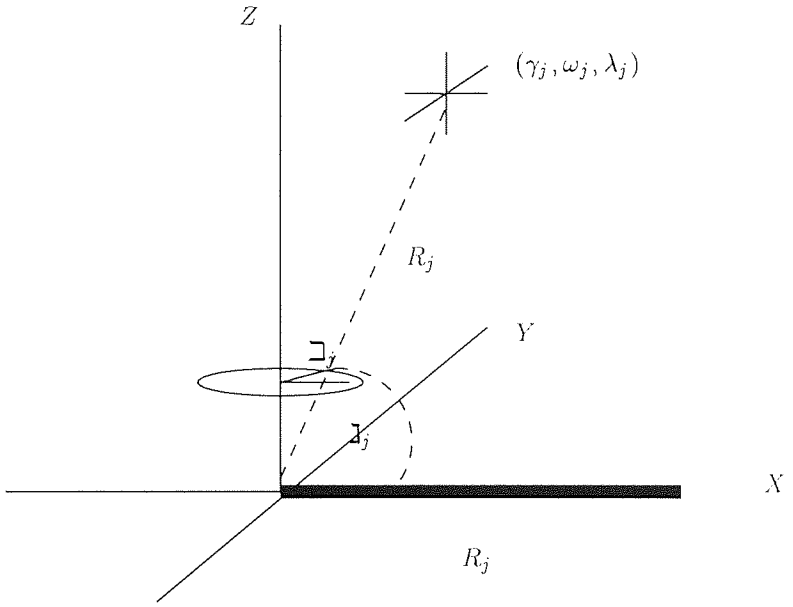

Figure 8.28: A Solution Group Factorisation of $\mathbb{R}^3$.

In this section, five canonical Lie factors and two canonical cyclic solution branch factors are determined for the general case of dextrous manipulators defined by $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathsf{SE}(3)$. As a particular case of this factorsation, four canonical Lie factors and one canonical cyclical solution factor are derived for the RR-K-1207 and Hollerbach manipulators defined by $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathbb{R}^3 \times \mathsf{SO}(2)$.

Using similar arguments to those of §8.2 for factorisation of $\mathbb{R}^{+2}$ into $\mathbb{R} \times \mathsf{S}^1$, $\mathbb{R}^{+3}$ may be factored as the product $\mathbb{R}^+ \times \mathsf{S}^2$, i.e. a spherical coordinate system similar to the polar coordinate system of $\mathbb{R}^{+2}$. The spherical coordinatisation of a point P in a Cartesian coordinate system $\{X_0, Y_0, Z_0\}$ is represented by $(R, \daleth, \beth)$ where R is the length of the line segment joining P to the origin, $\daleth$ is the angle between the line segment and the X_0 axis and $\beth$ is the angle between the line segment and the Y_0 axis.

Five representations of of SO(3) - fixed angles, Euler angles, equivalent angle-axis conventions, Quaternions and rotation matrices (see §7.4 and §2.1) - were analysed to determine the canonical solution group operator representation. The possible representations of SE(3) are these five representations of orientation times the Cartesian or spherical representation of $\mathbb{R}^{+3}$. A geometric analysis of these ten representations of SE(3) determined that the canonical solution group representation for SE(3) acts on the spherical coordinate representation of $\mathbb{R}^{+3}$ and the fixed angle representation of SO(3). The coupling of Cartesian orientation yaw λ and pitch ω coordinates to manipulator Cartesian position coordinates $(R, \daleth, \beth)$ are canonically represented in a spherical coordinate system of $\mathbb{R}^{+3}$ and the fixed angle coordinate system of SO(3). Since this section derives the optimal factorisation $\mathsf{G}_k, k = 1, \ldots 6$ as defined by (7.26) of solution group operators in the canonical representation, in the remainder of this section, the $r*$ prefix superscript, for instance on the solution group operators ${}^{r*}\Psi_k^{s*}, {}^{r*}\Phi_k^{s*}$, denotes the spherical coordinatisation of $\mathbb{R}^3$ and the fixed angle coordinatisation of O(3) and the $s*$ prefix subscript denotes the optimal derivation of solution group factors in this

[20] To observe this coupling, point your arm stright in front of you perpendicular to your torso and note the position of the centre point of your wrist. Determine the orientation of an orientation frame of your wrist relative to a fixed angle orientation coordinate system attached to a rectangular room corner. Now move your elbow 90^0 upwards to change the position of your wrist. The orientation of your wrist orientation frame relative to the room fixed angle orientation coordinate system will also been transformed by the elbow movement to reposition the wrist. Now move your wrist so that your hand is perpendicular to your forearm, i.e. change your wrist pitch orientation relative to the room orientation coordinate system. The position of the wrist is not changed by this change in wrist orientation.

representation.

If $\{X_j, Y_j, Z_j\}$, $\{X_i, Y_i, Z_i\}$ are orthogonal Cartesian coordinate frames, then the fixed angle orientation of $\{X_j, Y_j, Z_j\}$ with respect to $\{X_i, Y_i, Z_i\}$ is specified by successive rotations $\gamma_{j,i}$ around X_j to X_i, $\omega_{j,i}$ around Y_j to Y_i, and $\lambda_{j,i}$ around Z_j to Z_i. $\gamma_{j,i}$ is a roll transformation angle. $\omega_{j,i}$ is a pitch transformation angle. $\lambda_{j,i}$ is a yaw transformation angle. The application of $(\gamma_{j,i}, \omega_{j,i}, \lambda_{j,i})$ in the order left to right is important since the order does not commute. For 7 *dof* manipulators, let $\{X_0, Y_0, Z_0\}$ denote the base frame, and $\{X_7, Y_7, Z_7\}$ denote the wrist frame. $(\gamma_{0,7}, \omega_{0,7}, \lambda_{0,7})$ transforms $\{X_0, Y_0, Z_0\}$ to $\{X_7, Y_7, Z_7\}$. Conversely $(\gamma_{7,0}, \omega_{7,0}, \lambda_{7,0})$ transforms $\{X_7, Y_7, Z_7\}$ to $\{X_0, Y_0, Z_0\}$. $\gamma_{0,7}, \omega_{0,7}, \lambda_{0,7}$ represent the orientation of $\{X_7, Y_7, Z_7\}$ with respect to $\{X_0, Y_0, Z_0\}$.

The anthropomorphic Hollerbach and RR-K-1207 manipulators have an shoulder roll θ_1 and pitch θ_2, elbow roll θ_3 and pitch θ_4, wrist roll θ_5 and pitch θ_6 and tool plate roll θ_7 joints. Since they have no yaw joints, these manipulators cannot be arbitrarily Cartesian yaw oriented. For a given $(R_j, \beth_j, \beta_j)$ position of the RR-K-1207 and Hollerbach wrist, λ_j is constrained to be equal to $\beth_j$. This means in effect that the Cartesian space of these manipulators is in fact homotopic to $\mathbb{R}^{+3} \times \mathsf{SO}(2)$ with λ_j always constrained to be equal to $\beth_j$. We will however proceed with a general analysis of factorisation of $\mathcal{F} : \mathsf{T}^7 \times \mathsf{SE}(3)$ since other dextrous manipulators, such as a primate arm, do have a free λ_k orientation, in order to construct a complete theory of representation of kinematic transformations in nervous systems in §9.4.

Let $(R_j, \beth_j, \beta_j, \gamma_j, \omega_j, \lambda_j)$ be a representation of the position and orientation of the manipulator wrist frame $\{X_7, Y_7, Z_7\}$ at any position and orientation j in $\mathfrak{W}$ with respect to $(R_0, \beth_0, \beta_0, \gamma_0, \omega_0, \lambda_0)$ corresponding to $\{X_0, Y_0, Z_0\}$. Since in the general case of factorsation of $\mathsf{SE}(3)$, (ω_j, λ_j) are coupled to $(R_j, \beth_j, \beta_j)$, a change in position $(R_j, \beth_j, \beta_j)$ to $(R_i, \beth_i, \beta_i)$ where $R_i = R_j, \beth_i = \beth_j + \Delta\beth, \beth_i = \beth_j + \Delta\beth$ by the transformation $\Delta\beth$ followed by $\Delta\beth$ will induce by isometry a corresponding transformation of $(\gamma_j, \omega_j, \lambda_j)$ to $(\gamma_j, \omega_h, \lambda_h)$ where $\omega_h = \omega_j + \Delta\beth$ and $\lambda_h = \lambda_j + \Delta\beth$. γ_j is unchanged by the transformation since orientation frame roll γ is not coupled to position nor vice versa. Let $\gamma_i = \gamma_j + \Delta\gamma$ be followed by $\omega_i = \omega_h + \Delta\omega$ which is again followed by $\lambda_i = \lambda_h + \Delta\lambda$. $(R, \beth_i, \beth_i)$ are unchanged by these transformations. The sequence of five transformations maps $(R_j, \beth_j, \beta_j, \gamma_j, \omega_j, \lambda_j)$ to $(R_j, \beth_i, \beta_i, \gamma_i, \omega_i, \lambda_i)$. Since the sequence order is important and the transformations are the actions of the five ${}^{r*}\Phi_k^{s*}$ canonical solution group operators that transform ${}^{r*}\mathbf{x}_j^s \in \mathsf{SE}(3)$ to ${}^{r*}\mathbf{x}_i^s \in \mathsf{SE}(3)$, e.g. $\beth_j = {}^{r*}\Phi_1^{s*}(\beth_j)$, they are expanded and listed by order of composition

$$
\begin{aligned}
{}^{r*}\Phi_1^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \beth_i &= \beth_j + \Delta\beth, && (8.22)\\
f_1 : \mathsf{S} \rightarrow \mathsf{S} : \omega_h &= \omega_j + \Delta\beth, && (8.23)\\
{}^{r*}\Phi_2^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \beth_i &= \beth_j + \Delta\beth, && (8.24)\\
f_2 : \mathsf{S} \rightarrow \mathsf{S} : \lambda_h &= \lambda_j + \Delta\beth, && (8.25)\\
{}^{r*}\Phi_3^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \gamma_i &= \gamma_j + \Delta\gamma, && (8.26)\\
{}^{r*}\Phi_4^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \omega_i &= \omega_j + \Delta\beth + \Delta\omega, && (8.27)
\end{aligned}
$$

And in the general case of orientable yaw

$$
{}^{r*}\Phi_5^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \lambda_i = \lambda_j + \Delta\beth + \Delta\lambda. \tag{8.28}
$$

With a direct inverse function Υ^{-1} restricted to $(R^+, \beth_i, \beta_i, \gamma_i, \omega_i, \lambda_i)$ and ζ a direct parameterisation of the solution element fibre $\boldsymbol{\theta}_i^{s*}$ at $\mathbf{x}_i^{s*}$

$$
(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i) = \Upsilon^{-1}(R_j, \beth_i, \beta_i, \gamma_i, \omega_i, \lambda_i, \zeta). \tag{8.29}
$$

This general factorisation of canonical solution group actions on SE(3) applies to primate dextrous arms as well as manipulator arms. The ${}^{r*}\Phi_5^{s*}$ factor is derived for yaw orientation as a general case since the RR-K-12-7 and Hollerbach arms do not require this factor since they do not have orientable λ. ${}^{r*}\Phi^{s*} = {}^{r*}\ \Phi_5^{s*}({}^{r*}\Phi_4^{s*}({}^{r*}\Phi_3^{s*}({}^{r*}\Phi_2^{s*}({}^{r*}\Phi_1^{s*}()))))$ is the total $\mathcal{F}$ image solution group operator action expansion ${}^{r*}\mathbf{x}_i^s = {}^{r*}\ \Phi^{s*}({}^{r*}\mathbf{x}_j^s), \forall j$.

Geometric analysis of the RR-K-1207 and Hollerbach arms determines the corresponding ${}_j^{r*}\Psi_k^{s*^{-1}}$ of (8.3). See Figure 8.26 and Figure 8.27. The RR-K-1207 and Hollerbach arms have four roll joints - a shoulder θ_1, elbow θ_3, a wrist θ_5 and a tool plate θ_7 roll joint. Orientation roll $\gamma = \theta_3 + \theta_5 + \theta_7$. The arms have three pitch joints - a shoulder θ_2, an elbow θ_4, and a wrist θ_6 pitch joint. Analysis of the manipulator geometry in Figure 8.26 and Figure 8.27 reveals that

$$\gamma_j = \theta_3^j + \theta_5^j + \theta_7^j, \tag{8.30}$$

$$\omega_j = \theta_2^j + \theta_4^j + \theta_6^j, \tag{8.31}$$

$$\lambda_j = \beth_j. \tag{8.32}$$

If in the general SE(3) dextrous manipulator case (and the particular $\mathbb{R}^3 \times \mathbb{SO}(2)$ RR-K-1207 or Hollerbach arm case), a manipulator is at position and orientation $(R_j, \daleth_j, \beta_j, \gamma_j, \omega_j, \lambda_j)$ at $(\theta_1^j, \theta_2^j, \theta_3^j, \theta_4^j, \theta_5^j, \theta_6^j, \theta_7^j)$

1. a transformation $\theta_2^i = \theta_2^j + \Delta\theta_2$ of the shoulder pitch joint induces a transformation of the position $\daleth_j$ to $\daleth_i = \daleth_j + \Delta\daleth$, and orientation pitch ω_j to $\omega_i = \omega_j + \Delta\daleth$. By isometry,

$$\Delta\daleth = \Delta\theta_2. \tag{8.33}$$

Hence

$${}^{r*}\Psi_1^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_2^i = \theta_2^j + \Delta\daleth, \tag{8.34}$$

$${}^{r*}\Psi_1^{s*^{-1}} : \mathsf{S} \rightarrow \mathsf{S} : \theta_2^j = \theta_2^i - \Delta\daleth. \tag{8.35}$$

2. a transformation $\theta_1^i = \theta_1^j + \Delta\theta_1$ of the shoulder roll joint induces a transformation of the position $\beth_j$ to $\beth_i = \beth_j + \Delta\beth$ and the orientation yaw λ_j to $\lambda_i = \lambda_j + \Delta\beth$. By isometry

$$\Delta\beth = \Delta\theta_1. \tag{8.36}$$

Hence

$${}^{r*}\Psi_2^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_1^i = \theta_1^j + \Delta\beth \tag{8.37}$$

$${}^{r*}\Psi_2^{s*^{-1}} : \mathsf{S} \rightarrow \mathsf{S} : \theta_1^j = \theta_1^i - \Delta\beth \tag{8.38}$$

3. a transformation $\theta_7^i = \theta_7^j + \Delta\theta_7$ of the tool plate roll joint induces a transformation of orientation roll γ_j to $\gamma_i = \gamma_j + \Delta\gamma$ where $\Delta\gamma = \Delta\theta_7$. Hence

$${}^{r*}\Psi_3^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_7^i = \theta_7^j + \Delta\gamma, \tag{8.39}$$

$${}^{r*}\Psi_3^{s*^{-1}} : \mathsf{S} \rightarrow \mathsf{S} : \theta_7^j = \theta_7^i - \Delta\gamma. \tag{8.40}$$

4. a transformation $\theta_6^h = \theta_6^j + \Delta\theta_6$ of the wrist pitch joint induces a transformation of orientation pitch ω_j to $\omega_h = \omega_j + \Delta\omega$ where $\Delta\omega = \Delta\theta_6$. A transformation $\theta_6^i = \theta_6^h + \Delta\theta_2$ defined in (8.27) induces a transformation of ω_h to $\omega_i = \omega_h + \Delta\daleth$ where $\Delta\daleth = \Delta\theta_2$. Hence

$${}^{r*}\Psi_4^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_6^i = \theta_6^j + \Delta\daleth + \Delta\omega, \tag{8.41}$$

$${}^{r*}\Psi_4^{s*^{-1}} : \mathsf{S} \rightarrow \mathsf{S} : \theta_6^j = \theta_6^i - \Delta\daleth - \Delta\omega. \tag{8.42}$$

5. And finally in the general case of a yaw orientable dextrous manipulator, assuming that θ_5 is the wrist yaw joint, a transformation $\theta_5^h = \theta_5^j + \Delta\theta_5$ of the wrist yaw joint induces a transformation of orientation yaw λ_j to $\lambda_h = \lambda_j + \Delta\lambda$ where $\Delta\lambda = \Delta\theta_5$. A transformation $\theta_5^i = \theta_5^h + \Delta\theta_1$ defined in (8.36) induces a transformation of λ_h to $\lambda_i = \lambda_h + \Delta\beth$ where $\Delta\beth = \Delta\theta_1$. Hence

$$^{r*}\Psi_5^{s*} : \mathsf{S} \rightarrow \mathsf{S} : \theta_5^i = \theta_5^j + \Delta\beth + \Delta\lambda, \tag{8.43}$$

$$^{r*}\Psi_5^{s*^{-1}} : \mathsf{S} \rightarrow \mathsf{S} : \theta_5^j = \theta_5^i - \Delta\beth - \Delta\lambda. \tag{8.44}$$

$^{r*}\Psi^{s*^{-1}} = {}^{r*}\Psi_1^{s*^{-1}}(^{r*}\Psi_2^{s*^{-1}}(^{r*}\Psi_3^{s*^{-1}}(^{r*}\Psi_4^{s*^{-1}}(^{r*}\Psi_5^{s*^{-1}}())))))$ is the total $\mathcal{F}$ pre-image solution group inverse operator. $^{r*}\boldsymbol{\theta}_j^s = {}^{r*}\Psi^{s*^{-1}}(^{r*}\boldsymbol{\theta}_i^s)$.

In the general dextrous manipulator case $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathsf{SE}(3)$, the canonical solution group factor operators $^{r*}\Psi_k^{s*}, ^{r*}\Phi_k^{s*}, k = 1, \ldots 5$ generate the following factorisation of $\mathsf{L} = (\mathsf{T}^7, \mathbb{R}^{+3} \times \mathsf{SO}(3))$ into a quotient solution element manifold homotopic to

$$(\mathsf{T}^2 = \mathsf{T}^7/\mathsf{T}^5, \mathbb{R}^+ = (\mathbb{R}^{+3}/\mathsf{SO}(2)) \times (\mathsf{SO}(3)/\mathsf{SO}(3))). \tag{8.45}$$

Since the T^2 solution element submanifold is embedded in T^7, the solution element homotopic to $(\mathsf{T}^2, \mathbb{R}^+)$ can be directly parameterised by Υ^{-1} with a one dimensional ζ restricted parameterisation for instance homotopic to S

$$\Upsilon^{-1} : \mathbb{R}^+ \times \mathsf{S} \rightarrow \mathsf{T}^7 : (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i) = \Upsilon^{-1}(R_i, \zeta). \tag{8.46}$$

In the specific case of the RR-K-1207 and Hollerbach manipulators $\mathcal{F} : \mathsf{T}^7 \rightarrow \mathbb{R}^3 \times \mathsf{SO}(2)$, the canonical solution group factor operators $^{r*}\Psi_k^{s*}, ^{r*}\Phi_k^{s*}, k = 1, \ldots 4$ generate the following factorisation of $\mathsf{L} = (\mathsf{T}^7, \mathbb{R}^3 \times \mathsf{SO}(2))$ into a quotient solution element manifold

$$(\mathsf{T}^3 = \mathsf{T}^7/\mathsf{T}^4, \mathbb{R}^+ = (\mathbb{R}^{+3}/\mathsf{SO}(2)) \times (\mathsf{SO}(2)/\mathsf{SO}(2))). \tag{8.47}$$

The solution element homotopic to $\mathsf{T}^3, \mathbb{R}^+$ can be directly parameterised by Υ^{-1} with a two dimensional ζ restricted parameterisation for instance homotopic to T^2

$$\Upsilon^{-1} : \mathbb{R}^+ \times \mathsf{T}^2 \rightarrow \mathsf{T}^7 : (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i) = \Upsilon^{-1}(R_i, \zeta). \tag{8.48}$$

In both cases, (7.26) is minimised, i.e. both factorisations are $L_{\min}$. The geometric analysis of canonical solution groups of dextrous manipulator kinematics derived from first principles provides a foundation for specification of a fast solution method for inverse kinematics of dextrous manipulators. The problem of approximation of the non-linearity of $\mathcal{F}^{-1}$ has effectively been compressed from six (five) dimensions to a one dimensional line by the solution group analysis. Before analysis of the local linear approximation of the solution element defined on this line and parameterised by ζ, in order to define the restriction of ζ, it is necessary to analyse the homotopy structures of Υ in $\mathfrak{W}_w$ subsets of the line in order to determine where the solution fibre is contractabe and where it is disjoint and if disjoint over what embedding dimensions of T^7 is the fibre non-contractable.

8.3.2 The Homotopy of the Trivial Fibre Bundle Solution Element

The inverse solution fibre $\mathfrak{F}$ of a dextrous manipulator is homotopic to S^1 embedded in T^7 while that of the RR-K-1207 and Hollerbach arms is homotopic to T^2 also embedded in T^7. Like $\mathfrak{F}$ of Planar 3-$\mathbf{R}$, the solution fibre is contractable or non-contractable disjoint in different $\mathfrak{W}_w$. The global fibre bundle homotopy structure of full dextrous manipulators and the RR-K-1207 and Hollerbach arms is hoever

more complicated than that of planar 3-**R**. See (Kreutz-Delgado et al., 1992). At most locations in $\mathfrak{W}_w$, the Hollerbach and a RR-K-1207 manipulators have eight solution branches generated by the following solution branch factors:

1. wrist flip and wrist no flip solutions,
2. elbow up and elbow down solutions are seperated by the shoulder pitch joint,
3. elbow right and an elbow left (relative to to the XZ plane through the shoulder joint) solutions seperated by the elbow roll joint,

A sixth and seventh solution group are generated for the first and second cases. ${}^{r*}\Phi_6^{s*} = {}^{r*}\Phi_7^{s*} = \mathbf{I}$. For ${}^{r*}\Psi_6^{s*}$ in the case of general dextrous manipulators and primates that have a wrist that intersects in a point, the wrist orientation joints governing roll, e.g. θ_7,, pitch, e.g. θ_6 and yaw, e.g. θ_5, generate a canonical C_2 branch solution group with the fixed angle representation of orientation. For each orientation $(\gamma_j, \omega_j, \lambda_j)$, there are two wrist solutions $(\theta_5^1, \theta_6^1, \theta_7^1), (\theta_5^2, \theta_6^2, \theta_7^2)$ to reach $(\gamma_j, \omega_j, \lambda_j)$ with $\theta_5^2 = \theta_5^1 + \pi, \theta_6^2 = -\theta_6^1, \theta_7^2 = \theta_7^1 + \pi$. See (Craig, 1989; Fu et al., 1987) pg. 136.

For ${}^{r*}\Phi_7^{s*}$, the solution operator is similar to the branch operator derived for planar 3-**R** in (8.16) and (8.17) with θ_2 and θ_4 of the RR-K-1207 replacing θ_1 and θ_2 respectively of planar 3-**R**.

The number of solution branches at all points in every $\mathfrak{W}_w$, all singular configurations and all possible pre-image manifold parameterisations of a RR-K-1207 have not yet been derived (Kreutz-Delgado et al., 1992). (Kreutz-Delgado et al., 1992) analyse elbow, shoulder, wrist-wrist, and wrist-shoulder joint $\mathbb{CPS}$ for the zero offset Hollerbach arm and the RR-K-1207. Singularity analysis is complicated by non-zero offset parameters. Geometric analysis of the RR-K-1207 may eventually yield a complete description of its singularity and homotopy structures (Kreutz-Delgado et al., 1992). In the absence of this analysis of the homotopy of the fibre over each $\mathfrak{W}_w$ of the RR-K-1207, the homotopy determination search over ${}_d\theta^{\zeta-}$ parameterisations for each dimension d of the embedding space homotopic to T^n along which $\mathfrak{F}$ is non-contractable defined in §8.1.3.1 on p.g. 182 could determine the global solution element fibre homotopy structure of the RR-K-1207.

To recap, for each dimension d, where ${}_d\theta^{\zeta-}$ is the parameterisation of the fibre, after selection of arbitrary ${}^{r*}_0\boldsymbol{\theta}_i^s$, differential inversion of $({}^{r*}\mathbf{x}_i^s, {}^q_d\theta^{\zeta-} = {}^1_d\theta^{\zeta-} + q\Delta)$ by $\mathbf{J}^{\Diamond^{-1}}$ will generate a q solution ${}^{r*}_q\boldsymbol{\theta}_i^s$. If ${}^{r*}_q\boldsymbol{\theta}_i^s$ are defined for ${}_d\theta^{\zeta-} = 0, \ldots, 2\pi$ for ${}_d\theta^{\zeta-}$ in dimension d of the embedding space, then the fibre is non-contractable over dimension d, i.e. it has disjoint homotopy. If ${}^{r*}_q\boldsymbol{\theta}_i^s$ are not defined for ${}_d\theta^{\zeta-} = 0, \ldots, 2\pi$ for ${}_d\theta^{\zeta-}$ for any d, then the fibre is contactable.

For a disjoint fibre, the dimension d of the embedding space where the solution branch fibres are disjoint can be determined by a combinatoric search of the initialisations $\{{}^{r*}_0\boldsymbol{\theta}_i^s\}$ of $\mathbf{J}^{\Diamond^{-1}}$ inversion of $({}^{r*}\mathbf{x}_i^s, {}^1\theta^{\zeta-})$. $\{{}^{r*}_0\boldsymbol{\theta}_i^s\}$ can be generated by adding π to each of the coordinates of ${}^{r*}_1\boldsymbol{\theta}_i^s$. Dimension d is the disjoint fibre seperation dimension if the $\mathbf{J}^{\Diamond^{-1}}$ solution converges to ${}^{r*}_1\boldsymbol{\theta}^s_{i_d} \neq {}^{r*}_1\boldsymbol{\theta}_i^s$.

8.3.3 A Local Linear Approximation of the Solution Element

For construction of F^{-1}, θ^ζ those dimensions of the fibre embedding space that parameterisation of $\mathfrak{F}$ need to be selected. As mentioned in §8.1.3, for parameterisation of a disjoint $\mathfrak{F}, \theta^\zeta = \theta_{m+1}, \ldots, \theta_7$ are determined as those dimensions of the embedding space homotopic to T^7 along which $\mathfrak{F}$ is non-contractable and not those dimensions along which the fibre is disjoint. For contractable $\mathfrak{F}, \theta^\zeta = \theta_{m+1}, \ldots, \theta_7$ should be selected as those dimensions of the embedding space in which $\mathfrak{F}$ has the largest dynamic range since this makes the restriction of ζ_- easier to determine. Joints which satisfy

this condition are θ_7 and θ_5, the last two roll joints. The solution elment of the RR-K1207 homotopic to $(\mathsf{T}^3, \mathbb{R}^+$ can be directly parameterised with $\theta_5^{\zeta_{\leftharpoonup}}, \theta_7^{\zeta_{\leftharpoonup}}$ restricted parameterisation. See (8.48).

After definition of a tessellation density d, an orthogonal tessellation ${}^{r}_{\aleph}\mathbf{x}^{s}_{i}$ and a the restricted geodesic tessellation of ${}_{\aleph}\zeta_{\leftharpoonup}$ defined in §8.1.3.1, this induces the restricted tesselation ${}^{r}_{\sharp}\boldsymbol{\theta}^{\zeta_{\leftharpoonup}}_{i}$ by $\mathbf{J}^{\Diamond^{-1}}$ inversion of ${}^{r}_{\aleph}\mathbf{x}^{s}_{i}, {}_{\aleph}\zeta_{\leftharpoonup}$. See §8.1.3.1. ${}_{\aleph}\mathbf{J}^{\Diamond^{-1}}$ can be also evaluated at each f^{-1} tesselation point ${}^{r}_{\aleph}\mathbf{x}^{s}_{i}, {}_{\aleph}\zeta_{\leftharpoonup}$. In the case of the SE(3) dextrous and $\mathbb{R} \times$ SO(2) dextrous RR-K-1207 and Hollerbach manipulators, symbolic evaluation of ${}_{\aleph}\mathsf{J}^{\Diamond}$ is prohibitive so it is more effecient to evaluate and invert it numerically at each point of ${}^{r}_{\aleph}\mathbf{x}^{s}_{i}, {}_{\aleph}\zeta_{\leftharpoonup}$. See §8.1.3. f^{-1} is defined by $\{{}^{r}_{\aleph}\mathbf{x}^{s}_{i} \times \aleph\zeta_{\leftharpoonup}, {}_{\aleph}\mathsf{J}^{\Diamond^{-1}}\}$ and the local linear map approximation defined in §8.1.3.2. For a general ζ parameterisation, f^{-1} is defined on $\mathbb{R}^+ \times \mathsf{T}^2 \rightarrow \mathsf{T}^7$. However since $\theta_7^{\zeta_{\leftharpoonup}}$ and $\theta_5\zeta_{\leftharpoonup}$ are direct parameterisations of the solution element image of f^{-1}, definition of f^{-1} image can be reduced to T^5 to minimise storage complexity of f^{-1}. In addition the columns of ${}_{\aleph}\mathsf{J}^{\Diamond^{-1}}$ that are not functions of the f^{-1} pre-image variables $R, \theta_5\zeta_{\leftharpoonup}, \theta_7\zeta_{\leftharpoonup}$ do not need to be stored. Thus the storage cost for each element ${}^{r}_{\aleph}\mathbf{x}^{s}_{i} \times \aleph\zeta_{\leftharpoonup}, {}_{\aleph}\mathsf{J}^{\Diamond^{-1}}$ of f^{-1} of the RR-K-1208 and Hollerbach arms are $23 = 5+3+5\times3$ units of storage[21] per element. The evaluation of a f^{-1} image differential corresponding to a pre-image differenatial costs 15 operations.

An arbitary point in the solution element f^{-1} pre-image will be bounded by a cube in ${}^{r}_{\aleph}\mathbf{x}^{s}_{i} \times \aleph\zeta_{\leftharpoonup}$. The indices of the eight vertices of the bounding cube can be retrieved from a regular orthogonal tessellation by division of the components of the arbitrary point by the tesselation Δ along that component. The respective quotients and the quotients plus Δ gives the two cube bounds for the respective component. The bounding cube defined by these bounds can be determined by three more addition operations.

The cost of retrieval of ${}^{r}_{\sharp}\boldsymbol{\theta}^{\zeta_{\leftharpoonup}}_{i}, {}_{\aleph}\mathsf{J}^{\Diamond^{-1}}$ stored at each vertex, minimised by use of Hash coded addressing (see (Knuth, 1973)), can be bounded to $O(3d)$ where d is the maximum number of tesselation points in a dimension. The local linear approximation proposed in §8.1.3.2 involves the addition of the bounding cube vertex zero order terms and the first order terms after the evaluation of the first order term. This requires 8 addition operations and 8 subtraction operations in three dimensions, i.e. 48 operations, followed by 8×15 first order evaluation operations. These first order vetex approximations are finally combined together in five dimensions by 8 multiplication and 7 addition operations. Thus the local linear approximation at an arbitrary point in the solution element is of complexity $O(224 + 3d) = O(3 + 3 + 3 + 3d + 48 + 8 \times 15 + 8 \times 5 + 7)$. The fixed 224 arithmetic operations can be evaluated fast with a good arithmetic and logic unit. The main variable of solution complexity is d, the number of tesselation points per dimension. For an arbitary point in the total workspace, the cost of the solution group operations, i.e. 6 subtractions and 6 additions must must be added. Thus the total solution complexity is of $O(236 + 3d)$. The proposed[22] solution is thus as fast as the state of the art in RAM cache combined with an effecient hash addressing scheme. The f^{-1} approximation of the solution element could be compiled to a DSP chip for hard real-time operation.

(Kreutz-Delgado et al., 1992) propose a parameterisation of the redundancy of a RR-K-1207 by a scalar variable that is defined by the angle between the plane that passes through the shoulder θ_2, elbow θ_4 and wrist θ_6 joints, and the vertical plane through the X, Z axes. (Kreutz-Delgado et al., 1992) derive the functions that map the joint vector to this angle and augment the Jacobian with the derivative of these functions. (Kreutz-Delgado et al., 1992) analysis of the properties of the augmented

[21] For the general case, the storage required per element is $20 = 6 + 2 + 6 \times 2$ units per element.

[22] It would have made the thesis more complete to implement and test the solution for the RR-K1207. However, it was only after a substantial amount of time and effort had been wasted on unreported (because subsequently preempted) benchmarks of radial basis function and feed-forward network approximations of the solution element that it was realised that an analytical approximation was straightforward to derive but not to implement and validate. Hence implementation details are left as a future extension of this work.

Jacobian for iterative inversion reveal that the augmented Jacobian is singular when the arm is at a $\mathbb{CPS}$ or at a nonsingular configuration for which the arm angle ceases to parameterise the redundancy, i.e. at an arm angle representational singularity. The augmented Jacobian may be singular when the Jacobian is not not singular. (Kreutz-Delgado et al., 1992) conclude that unlike acyclic differential pseudoinversion of the Jacobian differential, augmented Jacobian inversion is cyclic over $\mathbb{CPS}$ free subspaces of the base space $\mathfrak{W}_w$ if the augmented Jacobian remains non-singular over $\mathfrak{W}_w$. F is thus also cyclic over $\mathbb{CPS}$ free subspaces of $\mathfrak{W}_w$ where ${}_{\aleph}\mathrm{J}^{\Diamond^{-1}}$ is non-singular. In addition by construction of F, where ${}_{\aleph}\mathrm{J}^{\Diamond^{-1}}$ is singular, a linear interpolation is used between elements of $\aleph\zeta_{\leftharpoonup}$ that maintains the solution cyclicity. In the multi-branch solution case, since branches are labelled by construction, a closed Cartesian trajectory which includes a traversal of a $\mathbb{CPS}$ is cyclic if it is inverted by F on the same approximated $\mathfrak{F}$ semifibres or solution branche after reverse traversal of the $\mathbb{CPS}$.

8.4 Optimisation of Inverse Kinematic Solution Group Trajectories

If a path planner computes a Cartesian path (e.g. Voronoi, see (Latombe, 1991)) $\mathbf{x}_t \in \mathsf{SE}(3), t = t_0, \ldots, t_g$ around obstacles bounded for instance by convex hulls in order for a manipulator to reach a target grasp position, there are infinitely many $\boldsymbol{\theta}_t$ trajectories to move a dextrous manipulator along the Voronoi path. With symmetry group operators ${}^{r*}\Phi^{s*}, {}^{r*}\Psi^{s*^{-1}}$ and a F^{-1} representation of the solution element, an inverse kinematic trajectory $\boldsymbol{\theta}_t$ can be generated that maximises an auxiliary path planning optimisation $O(\boldsymbol{\theta}_t)$ objective, for instance manipulability M, over the trajectory. The optimal integrated path is defined by

$$\max_{\zeta^t_{\leftharpoonup}} \int_{\mathbf{x}_{t_0}}^{\mathbf{x}_{t_g}} O({}^{r*}\Psi^{s*^{-1}}(F^{-1}({}^{r*}\Phi^{s*}(\mathbf{x}_t), \zeta^t_{\leftharpoonup}))) d\mathbf{x}. \tag{8.49}$$

If $O(\boldsymbol{\theta}_t) = M(\boldsymbol{\theta}_t) = \sqrt{|\mathbf{J}\mathbf{J}^T|}$, $\mathbf{J}$ is evaluated at $\mathcal{F}^{-1}(\mathbf{x}_t)$. The curve $\zeta^t_{\leftharpoonup}$ which maximises (8.49) may be determined for instance by ascent (initalised at the previous ${}^{\max}\zeta^{t-1}_{\leftharpoonup}$), in the $\frac{\partial O}{\partial \zeta^t_{\leftharpoonup}}$ direction to compute ${}^{\max}\zeta^t_{\leftharpoonup}$ that maximises M_t at $\mathbf{x}_t$. In the case of a manipulator with several solution branches, if a separate $F_1^{-1}, \ldots F_l^{-1}$ approximation of each solution branch is loaded onto seperate processors, a redundant inverse kinematic trajectory can be optimised over different solution branches in parallel. If the trajectory $\mathbf{x}_t$ is cyclic, i.e. $\mathbf{x}_{t_g} = \mathbf{x}_{t_0} \neq \mathbf{x}_t, t \neq t_0, t_g$, then the optimised O trajectory is also cyclic. (8.49) is maximised by a curve on a $\dim(\zeta)+1$ (time) dimensional F^{-1} pre-image manifold instead of a curve on the complete dextrous $7+1$ dimensional $\mathcal{F}^{-1}$ pre-image manifold which is the case if for instance the (Manocha and Canny, 1994) solution method or differential inversion of a pseudoinverse Jacobian determines $\mathcal{F}^{-1}(\mathbf{x}_t)$. Hence the individual solutions for $\mathbf{x}_t$ are computed faster and the optimisation problem is also potentially solved faster by the canonical solution group operation on local linear approximation of the solution element compared to differential pseudoinversion of the augmented Jacobian or iterative eigendcompositon of matrix polynomial solutions. The next Chapter proposes the solution developed in this Chapter as a basic model of inverse kinematic transformations encoded in primate nervous systems. The model replaces analytic determination of $\{{}^{r}_{\aleph}\mathbf{x}^{s}_{i}, {}_{\aleph}\zeta_{\leftharpoonup}, {}^{r}_{\sharp}\boldsymbol{\theta}^{\zeta^t_{\leftharpoonup}}_{i}, {}_{\aleph}\mathbf{J}^{\Diamond^{-1}}\}$ with iterative determination of these parameters by hierarchical extended topographic maps.

Chapter 9

A Hypothesis on Representation of Primate Arm Kinematics

Robot arms are usually rigid structures with a maximum $\frac{\text{payload}}{\text{manipulator}}$ mass ratio of $\frac{1}{10}$. Manipulator designs minimise link interaction forces that are not modeled by decoupled PID controllers. Human arms, on the other hand, are deformable structures that can be controlled with a $\frac{\text{payload}}{\text{manipulator}}$ mass ratio of the order of $\frac{10}{1}$. As well as these mechanical differences between robot and human arms, there are controller differences. Robot arms are controlled by digital computers. Human arms are controlled by nervous systems.

An adult human brain contains of the order of 10^{12} neurons and 10^{15} synapses. In $1mm^3$ of cortical tissue, there are approximately 10^5 neurons and 10^9 synapses. Each cortical neuron is connected to about 3% of the neurons in the surrounding square millimeter of cortex (Kandel and Schwartz, 1985). An adult human eye contains of the order of 1 million rods and cones. Each cell performs approximately 10 sensory transductions per second (Kandel and Schwartz, 1985). The brain can be modeled as a system that maps encoded sensory inputs to motor outputs (McCulloch and Pitts, 1943; Wiener, 1948). Sensorimotor coordination may be modeled by transformations of encoded visual information from retina-topic, head centred, and shoulder centred input coordinate frames to an effector coordinate frame where joint trajectories and muscle control are specified. The human arm has the kinematic structure of a dextrous **3R-R-3R** chain. See §2.2. The human arm has a spherical shoulder joint, rotational elbow joint, and a spherical wrist joint. The inverse kinematics of the human arm is under-determined. Control of the non-rigid deformable human arm is an even more highly under-determined inverse problem when inverse dynamic control is considered. The output dimensionality of control of human arm dynamics is of the order of several tens of thousand muscle signal channels.

9.1 Brain Modules Involved in Sensorimotor Transformations

The main human visuo-motor subsystems which are modeled in sensorimotor coordination theory are

1. visual sub-systems:
   (a) the lateral geniculate nucleus,
   (b) the superior colliculus,
   (c) the temporal lobe,

(d) the occipital lobe,

(e) the parietal lobe.

2. motor sub-systems:

   (a) the basal ganglia,

   (b) the motor cortex,

   (c) the pre-motor area,

   (d) the cerebellum.

See Figure 9.1. Modular models of visual information processing have been proposed for retina-topic maps that extract attributes such as shape, shading, colour, texture, and motion gradients from an image (Caelli, 1985; Marr, 1982). (Marr, 1982) modeled retinal transduction of photons into electrical impulses and feature detection by convolution filters of image intensity histograms which output a $2\frac{1}{2}$D primal sketch of chained detected lines. After image processing to striate cortex level, visual input channels fork into two distinct processing pathways (Rueckl et al., 1990):

1. **Inferio-temporal** cortex classifies **what** an object is.

2. **Posterior Parietal** cortex determines **where** an object is located.

The Posterior Parietal cortex is the most likely principal nervous site of kinematic transformations between head centred coordinate systems and limb centred coordinate systems. Posterior Parietal cells receive visual, auditory, somatic and vestibular sensory inputs; occulo-motor, head, limb and body afferent motor signals; and strong motivational projections from the limbic system. Parietal cell electrical discharge increases when an animal moves towards a sensed object, and also when it directs its attention to it. Posterior Parietal cells operate in a distributed manner to map sensory vector sets to other sensory reference frames or onto various motor coordinate systems (Berthoz, 1991). The Posterior Parietal cortex is the most probable site for integration of egocentric spatial relationships. An egocentric coordinate system is approximately equivalent to the Cartesian space of a manipulator.

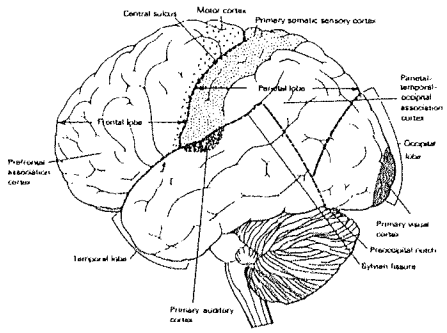

Figure 9.1: A Lateral View of the Major Modules of the Human Cortex. From (Kelley, 1985).

An other important brain system involved in sensorimotor coordination is the cerebellum. The cerebellum is a region of brain tissue that is composed of homogeneous highly regular substructures, each involved in many different information processing tasks related to movement control (Marr, 1982). The cerebellum's main function seems to be table lookup approximation and interpolation of muscle dynamic control equilibrium states. The cerebellum bears some resemblance to an expansion of radial basis functions. See §5.1 and (Albus, 1975).

9.2 Coordinate Systems for the Representation of Primate Eye-Arm Coordination

One hypothesis on representations underlying learning of coordination of visually guided grasping based on experimental human arm trajectory data is that a good model should have similar represen-

tations to the standard classical manipulator inverse control pipelines of Figure 1.2 (Berthoz, 1991). In this model, arm movement trajectories are initially modeled in kinematic frames, followed by an inverse transformation to movement dynamics. An extension of this general hypothesis, the lambda-hypothesis (Feldman, 1986), is that invertebrate inverse dynamic trajectories are best modeled as flexor-extensor muscle equilibria goals ω^g rather than as explicit torques τ^g. Flexor and extensor muscles about a joint θ can be modeled by a pair of opposing springs whose equilibrium state ω determines the state of θ. In the lambda model arm movement dynamics are modeled by $\{\omega_i^g, \ldots, \omega_n^g\}$ equilibria trajectories determined by a central nervous system. The spinal cord model selects optimal flexor-extensor equilibrium solutions for locally optimal control of deformable muscle dynamics.

Another model of sensorimotor coordination is that coordination involves side-minimisation of time or energy during movement. According to this model, nervous tissue supports some representation for regularisation of solutions of the under-determined inverse kinematic mapping. In this model the reaching solution path $\mathbf{c}(t)$ on the arm position manifold SE(3) from initial point $\mathbf{a}$ to goal point $\mathbf{b}$ is determined that minimises energy consumption E over $\mathbf{x} \in (\mathbf{a}, \mathbf{b})$

$$\underset{\mathbf{c}(t)}{\operatorname{argmin}} \int_{\mathbf{c}(t)} E(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) d\mathbf{x}. \tag{9.1}$$

The experimental literature on arm movements contains several hypothetical models of trajectory planning in body-centred Cartesian reference frames or trajectory planning in joint coordinates reference frames. Each model implies a different number of inverse kinematic transformations have to be performed for an inverse kinematic trajectory. If trajectory planning is modeled in joint space, only initial and end points of a trajectory must be inverted. (Rizzolatti, 1991) argues that empirically observed Gaussian arm acceleration profiles which minimise jerk cannot be explained by models of straight-line Cartesian trajectories if a joint reference space is used for trajectory planning. Hence (Rizzolatti, 1991) concludes in favour of a model of trajectory planning in a Cartesian body-centred coordinate systems.

However (Hollerback, 1990) argues that empirically observed kinematic path invariances across point to point arm movements in a sagittal plane under variances in speed and load can be minimally modeled in a joint reference frame while maintaining Gaussian acceleration profiles if staggered joint interpolation is used. Staggered joint interpolation means that different joints start and stop moving at different times t. A set of time scaling factors $\{c_i^t\}$ uniformly compress or expand joint acceleration profiles to compensate for time delays. One unspecified aspect of the (Hollerback, 1990) model is how subjects determine which θ_i to stagger and by how much for a given task. Knowledge about joint staggering delay could be modeled by an adaptive component for approximation of visual or environmental contact constraints. If indexed by Cartesian end points, joint staggering parameters could be retrieved from a model of motor memory.

There is one case where staggered interpolation does not provide a good model of Cartesian straight line trajectories. During joint reversal, some joints temporarily reverse their movement direction. When joint reversal is required to realise a Cartesian straight line, staggered joint interpolation breaks down and and a curved motion results. During experiments (Hollerback, 1990), subjects almost never performed a joint reversal. This empirical data seems to the (Hollerback, 1990) support model of trajectory planning in joint space.

Cortical and basal ganglionic electrical activity during movement changes in parallel without a functional hierarchy of sequentially recruited centres. Empirical observation of single cell recordings of sets of neuronal electrical activities observed during reaching tasks show electrical activity to be widely distributed throughout the cortex (Kelley, 1985). Nuclear Magnetic Resonance Imaging and

Positron Emission Tomography of the cortex during sensorimotor and cognitive tasks also show that neuronal electrical activity is widely distributed over different regions (Kelley, 1985).

These observation combined with knowledge that no analytical inverse kinematic solutions exist for a redundant 7 *dof* arm support a nonparametric or semi-parametric model of inverse human kinematic transformations. An approximation of dextrous manipulator inverse kinematics based on solution group operators acting on a local linear representation of a solution element has properties that recommend it as a model of representation of inverse kinematics in primate nervous systems. Before these properties are examined, background theories of the role of groups in perceptual, cognitive and nervous systems are reviewed.

9.3 Perceptual, Cognitive and Neural Information Transformation Groups

It appears that natural evolution concurrently engineered animal sensor and effector system designs and relatively precise modular brain organisation of sensory and motoric subsystem synaptic interconnectivity morphogenic probabilities which are hierarchically speciation invariant at gross modular architectural levels (Berthoz, 1991). Empirical neurophysiological data accumulated on the structure and chemical operation of animal and insect brains. e.g. see (Arbib, 1995; Arbib, 1987) indicates that architectural overlap of organisation of vertebrate brains is correlated with classifications of evolutionary divergence. All vertebrate brains have hemispheric symmetry. Niche demands enhance evolution of information processing tissues related to different tasks in different species, for instance owl foveal and night vision.

Across species, cerebral information processing increases in complexity during transition from sensory and motor peripheries to higher coordinative and cognitive functional levels. Theory of higher levels functionalities is more incomplete than at the sensory transduction level but extensive details of mind brain coupling have been established empirically since (Penfield and Roberts, 1959). The functional relationship of cellular activities to higher level cognitive level processing at the level of visually guided reaching is complex. Cognitive scientists construct information processing models, for instance of object recognition, at an abstract level where a single parameter may represent information processing of 10^5 neurons. Neuro-cognitive theories map correlations of neuro-histological activities, at various temporal and spatial scales (e.g. brain modules), to higher level information processing theories underlying for instance problem solving, emotional judgement, language, visually guided and ballistic arm reaching. Large data sets of single cell recordings of cat and monkey neurons engaged for instance in relatively basic visual tracking tasks are however extremely difficult to interpret.

Neurophysiologists usually construct relatively detailed information processing models of small fragments of tissues. Construction of isolated sub-system models is a simpler task than construction of more global models of sub-system interaction and modulation by high level global regulatory mechanisms involved in reasoning, memory, attention, hormonal activity and motivation. Nonetheless, while there is substantial localisation of mental function in the brain, much of the connectivity of the brain is also long distance inter-module and this structural inter-neuron communication traffic make it difficult to segment observations of brain dynamics during task performance and map these to a modular brain models.

A theoretical problem arises when one wishes to make the nervous system activity models refer to or denote objects in the environment or goals of an organism (a system that has goals, for instance survival) with respect to its environment. The introduction of the organism concept moves the the-

oretical description from physics and chemistry to biology with animal psychology hovering in the background. The introduction of nervous system reference to external events introduces an important set of modelling assumptions about reference of nervous system electrical events. The reference is not transparent and necessarily observerless. An observer is required to map nervous system activity models to coherent references to environmental events, particularly phenomena, i.e. events which can be sensed in the perceptual spectrum of a given organism. Assuming for technological purposes that intentionality is transparent and that a transcendental scientific observer position is unproblematical, one may posit the presence of mathematical groups underlying sensorimotor, neural, and cognitive information processing.

Symbolic cognitive psychology and neurophysiology are traditionally relatively autonomous inquiries because of the theoretical gulf seperating symbolic theories of symbolic thought from symbolic theories of neurophysiology. Since academic specialisation can result in similar ideas occuring in different fields (Foucault, 1980; Lyotard, 1984), an important goal of a multi-disciplinary study such as this is to attempt to systematise ideas common to different disciplines into bridging theories between disciplines.

Several researchers, for instance (Hoffman, 1966; Piaget, 1972; Edelman, 1989), have argued that transformation groups play a fundamental role in visual pattern recognition and sensorimotor coordination. (Hoffman, 1966) proposed a theory that Lie groups form a basis for shape invariant object recognition in binocular vision. The (Hoffman, 1966) Lie transformation group theory of vision is that the "the perceived world" has the structure of visual fields, defined on a Riemannian manifold, that represent contrast, spatial frequencies and positions of objects. (Hoffman, 1966) hypothesises that coordinatisation of object orientation, spatial and temporal frequencies, direction, and motion parameters is defined in differential space-time coordinate systems. Differential parameters are represented as vector fields and Lie algebra of fields is defined as in §7.3. In the (Hoffman, 1966) model, as visual information is processed hierarchically, successive levels of the visual cortex add to the visual manifold small direction fields at each point on the manifold thus filling out the objects orientation, spatial and temporal frequencies, direction, and motion vector field descriptions. The (Hoffman, 1966) theory defines a Lie algebra of operators for affine translations, time translations, rotations, hyperbolic rotations, a dilation group, and a Lorentz group of motion invariance of the various visual fields. A Lorentz transform is a transformation between two moving coordinate systems observed in a third coordinate system. (Hoffman, 1966) uses the Lorentz transform to define binocular vision motion invariance. The visual cortex has a hierarchy of cells that are driven by retinal ganglion cells and affected by the functions of neighbouring cortical cells. There is some experimental evidence that input signals to the hypercomplex cells are differential in form, i.e. cells communicate signal gradients underlying the construction of the (Hoffman, 1966) visual field theory.

Groups also play an important role in (Piaget, 1972)'s genetic epistemology. (Piaget, 1972) predicates development of higher cognitive functioning such as symbolic language and abstract reasoning on correct development of sensorimotor driven categorisation processes. According to (Piaget, 1972), a child must abstract group invariances of sensorimotor experiences, for instance the concept of number and volume invariance, in order to progress from sensorimotor coordination to operational, linguistic and mathematical reasoning

> "... concrete action precedes and makes possible the use of intellect ... As the infant begins to manipulate the objects which surround him, he gradually develops a practical 'understanding' of external reality ... Thus the acquisitions of the sensorimotor period form the foundations of the individual's mental development ..." ((Ginsburg and Opper, 1969), p. 106)

At the neurophysiological level, (Edelman, 1989) argues that the development of sensorimotor coordination at the kinematic level is coupled to acquisition of categorical primitives. As they interact with signals from sensory sheets, motor ensembles provide a nervous system with a mechanism for honing sensory feature correlation. Based on feature detection, categorisation occurs, in parallel and motor driven, of topological group invariances and continuities that are crucial for detailed sensory abstractions

> "... action is fundamental to perception, and sensory sheets and motor ensembles must operate together to yield a sufficient basis for perceptual categorisation ... The major and essential contribution of motor ensembles to perception is feature correlation, which arises out of the continuity properties of motion and the continual focusing of sensory signals by creating postural and gestural movements." ((Edelman, 1989), p.238)

(Edelman, 1989)'s theory is related to (Gibson, 1968; Gibson, 1979)'s concept that object representations include a set of functional affordances. These are the set of sensori-motor interaction possibilities of an object which have been incorporated into object representations during sensorimotor development.

Groups provide a theoretical mechanism for unifying many branches of mathematics (Weyl, 1952; Weyl, 1946). (Edelman, 1989; Hoffman, 1966; Piaget, 1972)'s group theories of the development of sensorimotor, perceptual and cognitive groups seem plausible in isolation. Considered together, they would seem to imply that a model could perhaps be generated of hierarchical groups that structure information representation and processing at neurophysiological, perceptual and cognitive levels. The inverse kinematic transformation of dextrous arms has an interesting group structure, is close to the nervous level of representation and it forms a basis for specification of perceptual invariances that underlie robust action in a wide variety of environments. Hence it is likely that the representation of the inverse kinematic transformation in nervous systems plays a key role in structuring the linkage of perception to action. Such a representation may also lay the foundations for higher level cognitive symbolic reasoning underlying strategic and tactical problem solving to maximise the primary primate behavioural objective function - survival.

9.4 A Hypothesis on the Nervous Representation of Inverse Kinematic Transformations

The approximation of inverse kinematics of dextrous manipulators developed in Chapter 8 has two basic components

1. a set of inverse kinematic solution group operators (Ψ_j, Φ_j).
2. a local linear network representation $\mathcal{F}_i$ of the solution element $\mathbf{y}_i = (\boldsymbol{\theta}_i, \mathbf{x}_i)$ such that

$$\Upsilon^{-1} : \mathbb{R}^+ \times \mathsf{S} \rightarrow \mathsf{T}^7 : (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i) = \Upsilon^{-1}(R_i, \zeta_j). \tag{9.2}$$

   See (8.46).

The local linear network is composed of $(R_i, \zeta_j, \theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i, \mathbf{J}_{i,j}^{\Phi^{-1}})$ for each element i, j, indexes of a regular orthonormal tessellation of $\mathbb{R}^+ \times \mathsf{S}$.

It is not necessary that a DH coordinatisation of links is represented in nervous systems. Any coordinate system for joint angles is appropriate that supports definition of of an augmented Jacobian inverse approximation of the solution element tangent bundle. If it is assumed that a primate

nervous system uses a fixed angle representation of orientation relative to a head or shoulder centred coordinate system, then a solution group and a local linear approximation of a solution element can model representation of inverse kinematics in primate nervous systems. Some modifications of the approximation are required to convert it to a model of inverse kinematics in nervous systems. In the previous chapter, a regular hyper-cube tessellation of the Cartesian subspace of the solution element was defined. At each lattice vertex coordinate (R_i, ζ_j), the corresponding restricted solution $(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i)$ and Jacobian approximation $\mathbf{J}_{i,j}^{\Diamond^{-1}}$ are stored to generate the combined interpolant and Jacobian function approximation around the Voronoi cell defined for (R_i, ζ_j). See §8.2.3. While it is plausable that $\mathbf{J}_{i,j}^{\Diamond^{-1}}$ could be represented in liminar cortical sheets, a theory is required to explain how a nervous system could learn to approximate $\mathbf{J}_{i,j}^{\Diamond^{-1}}$.

Such a theory is provided by a modification of the (Ritter et al., 1992) model. See §5.3.2. The (Ritter et al., 1992) model is a reasonable partial model of the development of a local linear map representation of inverse kinematics of primate arms. It is only a partial model since without the inclusion of a symmetry group analysis, it does not scale to modelling the complete dextrous primate arm homomorphism. The (Ritter et al., 1992) model maps visual measurements of feature positions in stereo-images directly to joint coordinates. See (Ritter et al., 1992) for a discussion of features required by stereo-images to measure orientation pitch and roll and a discussion of inversion of redundant mappings. The proposed model differs from the (Ritter et al., 1992) model with respect to the parameterisation of redundancy and with respect to the omission of a Cartesian space representation. The solution group operators act on a Cartesian subspace solution element. Hence a Cartesian space, corresponding to an egocentric head-centred coordinate system, is required by the proposed model. The (Ritter et al., 1992) algorithm is applied to determination of the local linear map elements $(R_i, \zeta_j, \theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i, \theta_7^i, \mathbf{J}_{i,j}^{\Diamond^{-1}})$. (Ritter et al., 1992) proposed that the local linear map is an approximation of a minimum norm pseudoinverse of the redundant Jacobian. In the proposed modification of the (Ritter et al., 1992) model, ζ_j parameterises solution redundancy. $\mathbf{J}^+$, the pseudoinverse Jacobian of (Ritter et al., 1992) is replaced by $\mathbf{J}_{i,j}^{\Diamond^{-1}}$.

Function approximation pre-image pre-processing is the usual way to reduce the **curse of dimensionality** (Bellman, 1961) of power-law sample scaling for uniform bound of the error of nonparametric multi-variate function approximation. To recap, the curse of dimensionality is that in general the number of samples required for well specified regression grows exponentially with the dimensionality d of the regression input space. If each variable is discretised into M measurements, scaling of regression is of the order of M^d. See (Bishop, 1995; Ripley, 1996). Solution group operators and a local linear representation of the solution element of a primate arm homomorphism are more efficient representations of the primate homomorphism than a complete nonparametric approximation based for instance on a (Ritter et al., 1992) fit to solution branches partitioned by methods of (DeMers, 1993). See §5.3.4. For a given discretisation level b of each pre-image dimension local linear basis functions:

1. the solution element approximation defined on the solution element subset of $\mathbb{R} \times \mathsf{S} \rightarrow \mathsf{T}^6$ requires $O(b^2)$ local linear elements. For $b = 10$, the number of basis functions required is approximately 100. For $b = 20$, the number of basis functions required is approximately 400.

2. a full nonparametric approximation defined on the workspace subspace of $\mathsf{SE}(3) \times \mathsf{S} \rightarrow \mathsf{T}^7$ requires $O(b^7)$ local linear elements. For $b = 10$, the number of basis functions required is approximately 10m. For $b = 20$, the number of basis functions required is approximately 1,280m.

The representation efficiency ratio r of local linear representation of the complete homomorphism to local linear representation of the solution element (for $b = 10, r = 100,000$; for $b = 20, r = 3,200,000$) is

actually higher because the local linear elements of the solution group have less parameters than those of a nonparametric approximation of the complete mapping. For approximation of the solution element, each linear basis function has $20 = 2+6+12 = \dim(R_i, \zeta_j)+\dim(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i)+\dim(\mathbf{J}_{i,j}^{\Diamond^{-1}})$ (if $\zeta_j = \theta_7^i$) parameters. For approximation of the complete mapping nonparametric elements, each linear basis function has $63 = 14+49 = \dim(\mathsf{T}^7)+\dim(\mathsf{SE}(3)\times\mathsf{T})+\dim([\mathsf{T}^7]^T[\mathsf{SE}(3)\times\mathsf{T}])$ parameters. At a discretisation level of $b = 10$, an assuming that each parameter requires floating point precision of 4 bytes, then the storage required for the local linear approximation of the solution element is approximately 8k bytes. The corresponding storage required by the local linear approximation of the complete mapping for equivalent approximation precision would require 2.52 Gigabytes. The local linear solution element approximation is of the order of a million times more storage effecient than a local linear nonparametric approximation of the complete mappong at discetisation level 10. These ratios demonstrate that the local linear representation of the solution element is five to seven orders of magnitude more representationally efficient than a local linear representation of the complete dextrous arm homomorphism depending on discretisation level b. Similar arguments hold for relative computational complexity ratio to determine the parameters of the basis functions by the (Ritter et al., 1992) algorithm for a local linear approximation of the solution element vs the complete homomorphism.

It is claimed that a combination of the solution group analysis and the analytic determination of the solution element with arbitrary precision over the solution accuracy is a better representation in terms of storage efficiency, computational complexity to determine parameters, and accuracy of the approximation (low bias and low estimator variance in terms of regression) than a non-parametric estimator of the dextrous manipulator inverse kinematic mapping assuming clustering heuristics (for instance those developed by (DeMers, 1993), see §5.3.4) are used to separate solution branches or by use of mixture density model objective functions. See (Bishop, 1995). The superiority of the solution group and analytically determined solution element over a full non-parametric approximation can be quantified:

1. One to ten million times more storage efficient.

2. At least several trillion times more computational complexity efficient since the determination of non-parametric regression parameters scales exponentially with dimensionality of the problem.

One of the aspects of the proposed modified (Ritter et al., 1992) model that bears some verisimilitude to the development of human arm coordination is tessellation of the solution element Cartesian subspace in proportion to the input probability density of calibration samples. See pg. 84. This agrees well with experimental data that practice of fine motor movements, for instance in sports, increases control accuracy of inverse kinematics of a human arm, presumably by increasing lattice density in areas of fine motor practice. However the actual tessellation topology of the (Ritter et al., 1992) model, a hyper-cubic mesh was perhaps motivated by ease of algorithm implementation, as in the case of the grid tessellation of the solution element approximation developed in Chapter 8, rather than possible correspondence with development of central nervous sheets. Efficiency of solution is a guiding force of design of nervous systems. This is a strong argument in favour of the solution group and local linear representation of the inverse kinematic solution element of primate are homomorphisms. Perhaps the nervous system represents the solution element subset of $\mathbb{R}\times\mathsf{S}$ with a Delaunay tessellation.

For approximation of functions with bounded second derivative such as the inverse kinematics of the solution element of a primate arm homomorphism, piecewise-linear approximations based on a Delaunay tessellation has a smaller worst case error than that based on hyper-cubic tessellation of solution element Cartesian subspace. A Delaunay tessellation is a simplex partition of a k-dimensional

Euclidean space where $k+1$ points generically determine a sphere and a primary k simplex. $(k+1)k$ edges emanate from each vertex denote topological neighbourhood relations between vertices. A set of $k+1$ node points form the vertices of a simplex in a Delaunay lattice if and only if a sphere which they determine does not contain any other node points. Bounding spheres of simplices in a Delaunay are as small as possible and triangles are as equilateral as possible.

In the previous chapter, is was shown that $\mathbb{R}^{+2} = \mathbb{R}^{+} \times \mathsf{S}$. A hexagonal lattice is the optimal Delaunay tessellation of $\mathbb{R}^2$. Thus a hexagonal Delaunay tessellation of $\mathbb{R}^{+2}$ will induce the corresponding Delaunay tessellation of $\mathbb{R}^{+} \times \mathsf{S}$. Thus the pre-image of the solution element of a primate[1] arm is defined on a disc in $\mathbb{R}^{+2}$ perhaps with a Delaunay tessellation of the disc. Such a hexagonal tessellation of a disc could be directly represented by a two dimensional nervous sheet in posterior parietal cortex.

The modified (Ritter et al., 1992) local linear approximation of a dextrous arm homomorphism solution element combined with the canonical solution group operators defined in Chapter 8 is proposed as a model of inverse kinematics in nervous systems. The representability of the solution group operators in the nervous system is not problematical[2] since they involve additions and subtractions of periodic variables to joint angles and orientation pitch, roll, and yaw parameters. It is hypothesised that solution group operators are represented in the posterior parietal cortex and the local linear approximation of the solution element is represented in the posterior parietal cortex or the cerebellum. The proposed model has several properties that recommend it as a model of kinematic transformation in primate nervous systems:

1. a reduction in consumption of nervous tissue for representation of kinematic mappings, i.e. efficiency of packing of information in nervous tissues, compared to complete nonparametric representation.

2. a reduction in computational complexity to determine parameters of the local linear basis functions compared to a complete nonparametric representation.

3. a fast solution. The speed of solution is bounded by the addressing speed of the nervous system.

4. a fast inversion of kinematic trajectories by traversal of group operators.

5. guaranteed cyclicity of inversion since the group structure and all singularities are explicitly labeled.

6. easy to implement control over singularity avoidance.

7. empirical evidence.

A hypothesis must be testable. The canonical solution groups in $\mathsf{SE}(3)$ are canonically represented in a fixed angle representation of $\mathsf{SO}(3)$ and a spherical coordinate representation of $\mathbb{R}^3$ attached to the shoulder. Hence this implies that this coordinate system is the shoulder representation system at least if not the head centred coordinate system. This can be tested by generation of an appropriate psychophysical and sensorimotor task experiment.

[1] It is more difficult to define the Delaunay tessellation of the $\mathbb{R}^{+} \times \mathsf{T}^2$ pre-image of the solution element of the RR-K1207 and Hollerbach arms. $\mathbb{R}^{+} \times \mathsf{T}^2$ is embedded in at least $\mathbb{R}^{+4}$. The Delaunay tessellation of higher dimensional Euclidean spaces is not a straightforward generalisation of hyper-hexagonal lattices. An octahedral lattice, the face-centred dual lattice of a cubic lattice, is the optimal Delaunay tessellation of $\mathbb{R}^3$ (see (Hilbert and Cohn-Vossen, 1932)). Specification of the Delaunay tessellation of $\mathbb{R}^{+} \times \mathsf{T}^2$ is left as an extension of this work.

[2] If it is assumed that a nervous system can represent periodic variables and algebra of periodic variables. This basic assumption is made by every mathematical model of kinematic transformation in nervous systems.

The presence of solution groups in the representation of the inverse kinematics of primates would be indicated if infants spent a large of amount of time approximating the fine control movement on the solution element. Such a control system would spend a large amount of time exploring the control vectors on the hexagonal tessellation of the solution element traversing $\mathbb{R}^+$ and ζ. Exclusive traversal of $\mathbb{R}^+$ and ζ is not empirically observed in infants. Rather infants traverse $\mathbb{R}^+$ and engage in primary, secondary, and tertiary circular motions of the wrists with visual monitoring of motions. See (Piaget, 1972). Infants may be engaged in a search of the Special Orthogonal Group of wrist orientations local to the solution element in addition to contraction on to an approximation of the solution element. If this is the case, it indicates that the solution group operator parameters are neither phylogenitically nor ontogenitically pre-coded. While it is argued that an infant eventually contracts onto the correct solution group parameters and an optimal Delaunay tessellation of the solution element for local linear approximation of Υ^{-1}, it seems likely that an infant determines this inverse kinematic solution representation via a simultaneous search of the solution element and the infinitesimal invariance of the solution group operators acting close to the solution element during its first weeks of sensorimotor activity. Simultaneous contraction on to a solution element approximation and determination of solution group parameters seems to be a difficult approximation problem if some form of the inverse kinematic solution group operators, at the very least the canonical coordinate systems, is not phylogenetically coded to restrict ontogenetic solution search.

The development of sensorimotor coordination provides a foundation of the development of number and continuity intuition. It is possible the the apprehension of number is allied to sensorimotor development, possibly even contingent on some ontogenetic sensorimotor paths. The development of intuition of number may a prerequisit for logical thought in so far as number distinguishes categories such as thinker, one, from the environmement, not one. At a basic level, apprehension of self seems to imply a concept of unity required for assertion of a Cartesian *cogito*. Large increase of robotic functionality, i.e. approaching the assembly capability of a four old child, may or may not require better,i.e. formal, understanding of development sequence of the unconscious substrate of mathematical reasoning.

Chapter 10

Conclusions

The research foreground reviewed, compared, and analysed the intersection of several background knowledge domains: functionalist and object oriented distribute software system engineering, solution groups of homomorphisms, manipulator inverse kinematics, model calibration, nonlinear nonparametric regression and sensorimotor coordination theory.

10.1 A Summary of the Foreground Research Results

The foreground research results are:

1. A theory of representation of kinematic transformations in primate nervous systems. See §9.4.
2. A hypothesis that the shoulder centred coordinate system and perhaps the head centred coordinate system is a product of spherical and fixed angled coordinate systems. See §9.4.
3. A review of the evidence for the presence of transformation groups in perceptual, cognitive and neural information processing systems. See §9.3.
4. An evaluation on theories of where and how inverse kinematic transformations may be represented in primate nervous systems. See §9.1 and §9.2.
5. An analysis of the the limits of robotic manipulator and automation systems based on an analysis of the limits of system software engineering - see Chapter 1, §1.1, §1.2 §5.4, §6.2, §6.3, §6.4, §6.6; and an analysis of the limits of understanding of sensori-cognitive-motor processes §9.3.
6. An analysis of the distributed computing standards that are likely to underlie next generation distributed automation systems. See §6.4, §6.5, and §6.6.
7. An analysis of solution groups of a manifold implicitly defined by a kinematic homomorphism. See §7.5 and §8.1.
8. A definition of optimality criteria that structure a geometric factorisation of canonical representations of solution groups of a homomorphism. See §7.7 and §8.1.1.
9. An elimination of algebraic factorisation of the canonical solution groups of 7 *dof* manipulators where the kinematic homomorphism is defined on a Denavit Hartenberg coordinate representation. Algebraic factorisation cannot determine the canonical representations of the homomorphsim pre-image and image and it is computationally intractable without a numerical su-

percomputer to determine the solution groups in the canonical representations once these have been algebraically derived. See §7.6.

10. An analysis of the homotopy structure of the trivial fibre bundle solution elements of canonical geometric factorisations of manipulator homomorphisms. See §8.1.2, §8.2.2 and §8.3.2.

11. A definition of canonical representations of solution groups of a Homomorphism from the n-Torus to the Special Euclidean Group. See §8.3.

12. A solution group factorisation of the Robotics Research K-1207 dextrous manipulator. See §8.3.

13. A definition of a constructive local linear approximation of the solution element of a canonical solution group of a dextrous manipulator homomorphism. The approximation is a combination of a local interpolation and a Taylor approximation and a definition of the error bounds is also derived. See §8.1.3.

14. A comparison of the error properties properties of the combined linear interpolant and Taylor approximtion with the error properties of higher order polynomial interpolation and bi-cubic interpolation. See §8.1.3.2.

15. An implementation of the combined linear interpolant and Taylor approximation for approximation of a solution branch of planar 3-**R** which is accurate to one minute of a degree. See §8.2.3.

16. A definition of a local linear network approximation of the solution element of the Robotics Research K-1207. See §8.3.3.

17. A proposed correction of the (Kreutz-Delgado et al., 1992) scalar parameterisation of the kinematic redundancy of the RR-K1207 arm to a 2 dimensional parameterisation. See §8.3.

18. A definition of a solution group factorisation of Planar 3-**R**. See §8.2.

19. An analytic definition and implementation verification of a local linear representation of the solution element of Planar 3-**R**. See §8.2.3.

20. An analytic solution of the inverse kinematics of planar 3-**R** based on an exponential representation of the Argand plane. See Appendix E.

21. A benchmark of nonparametric approximations of the inverse kinematics of planar 3-**R** that demonstrates the nonscalability of nonparametric approximation of homomorphisms between Lie groups without decomposition of the homomorphism into solution groups that act on a solution element. See Appendix F.

22. A derivation of the comparative integrated error scaling equations of nonparametric approximation of a homomorphism solution element vs nonparametric approximation of a complete homomorphism solution branch. (F.2), (F.3) were verified by an exhaustive cross validated benchmark on nonparametric approximation of inverse kinematics of planar 3-**R**.

23. A definition and analysis of error measures for simultaneous calibration of DH and stereo-camera parameters. See §4.1.

24. An analysis of equi-output parameter groups of the composition of stereo-camera and DH models. See §4.2.

25. A derivation, implementation and validation of simultaneous calibration of the composition of homogenous or pan-tilt stereo-camera and DH models. See §4.3.

26. A derivation, implementation and numerical experimental validation of optimal re-calibration of DH parameters with numerical products of DH matrix derivatives. See §3.3.

27. A specification and implementation of a distributed object environment for prototyping of manipulator control algorithms for physical manipulators and stereo-cameras. See §6.1 and Appendix C.

28. An extension of the matlab environment to control robot manipulator and image processing boards by a combination of `cmex` external functinon calls and sockets. See §6.1 and Appendix C.

29. A specification and implementation of a graphical tele-command interface to robots and cameras which supports point and click on stereo-image based manipulator kinematic command. See §6.1.4.

30. An analysis of the problems that matrix organisational structures cause in software engineering and a proposal for reduction of these problems. See §6.5.

31. An analysis of the limitations of functional and object oriented software engineering for scalable system integration. See §1.1, §6.2, §6.3, §6.6.

32. A methodology FOCH - Functional Object Composition Hierarchy, for integration of scalable dependable systems. See §6.3.

33. An integration of hierarchical system performance modelling into FOCH for prediction of system verification targets. See §6.5.3.

34. A formal definition of a software pattern. See §6.3.

35. A specification of a mapping from the Common Object Request Broker Architecture to the Open Distributed Processing Reference Model. See Appendix A.

36. A specification of requirements for Java-based open distributed agile manufacturing systems engineering. See §6.6.

37. A definition of core non-functional requirements for a virtual distributed kernel event transport system in §6.5 that will scale to support the outlined Java-ODP robotics and automation superstructure defined in §6.6.

38. A definition of required ODP manufacturing superstructural operations interfaces - secure real-time non-blocking nested two phase commit query transactions on replicated persistent objects; a transaction synchronised replica proxy object cache that can be updated by a dynamically created interfaces - to a distributed event transport system. See §6.5.4.

39. A definition of the tradeoffs in software engineering of integrated syste dependability and system scalability. See §6.5.2.

40. A review of software system integration methodologies and issues for development of manipulator control architectures. See §1.1, §6.2 §6.4, §6.5.4 and §6.6.

41. An outline of a software portability pattern for implementation of a portable micro-kernel. See §6.5.5.

42. An evaluation of the limited role of nonparametric regression and function approximation in control of manipulators at the kinematic (including path planning) level. See §1.3, §5.3, §5.4 and §5.4. Some indications of constraints on deployment of nonparametric controllers at dynamic levels are also indicated in §5.3 and §5.4.

43. A review and comparison of classical representations and solutions of inverse kinematics of non-redundant and redundant manipulators. See Chapter 2.

44. An implementation in C on UNIX and in parallel C on transputers of the (Ritter et al., 1992) hierarchical topographic map regression algorithm combined with numerical experiments that verified the (Ritter et al., 1992) results. See §5.3.2, pg. 86.

45. A derivation, implementation and validation of an algorithm for robust segmentation of a red LED image. See §6.1.2.

46. An analysis of the accuracy of image measurements of LED position. See §6.1.2 and §6.1.2.1.

The forty six research results may be classified into four categories. The final four sections of the thesis discuss these result categories and propose future research extensions of each category.

10.2 Sequential and Simultaneous Calibration of Cameras and Manipulators

For acceleration of exact Newton second-order re-calibration of DH parameters after a sequential calibration of stereo-camera and DH parameters, an optimal numerical evaluation of DH matrix first order and second order error derivatives with respect to a re-calibration error function was derived, implemented and tested in Chapter 3. The derivation of an efficient numerical evaluation of homogeneous transform products and derivatives for direct re-calibration of DH parameters in Chapter 3 is simple and more efficient than (Zhang and Roth, 1993)'s solution. Efficiency of numerical methods for re-calibration of parametric models is important since parametric models are highly nonlinear multi-dimensional functions. Implementation and validation of the algorithm indicates that it results in recalibration that is as accurate as end effector sensor and joint encoder measurement errors and part machining errors. Since the number of additions and multiplications required to determine DH parameters has been reduced to a minimum, no extensions of this method are possible and it is concluded that the recalibration method is suitable for implementation on cheap hardware.

Simultaneous calibration of a composition of stereo-camera and manipulator kinematic models is under-determined by equi-output parameter groups in the composition of stereo-camera and Denavit Hartenberg (DH) models. Simultaneous model calibration constraints were identified in Chapter 4. An error measure for simultaneous calibration of a composition of models was derived and parameter subsets with no equi-output groups were determined by numerical experiments to simultaneously calibrate the composition of homogenous or pan-tilt stereo-camera with DH models. Simultaneous calibration of combined models has also been analysed by (Zhuang et al., 1993; Bennet et al., 1991; Mooring et al., 1991). (Bennet et al., 1991)'s analysis of simultaneous calibration of a planar manipulator and planar stereo-cameras was based on a calibration error measure similar to (4.2). Chapter 4 analysed full pan-tilt stereo-cameras and 3 and 6 *dof* manipulator calibration. A calibration error

function was specified in stereo-image space in order to avoid estimation of propagation of input noise through camera models to Cartesian space. The proposed combined calibration method could be extended with:

1. **either** an optimal numerical design (see (Box et al., 1978)) to determine all of the subsets of the combined model which when fixed induce a null equi-output transformation group on the complement parameter sets **or**

2. determination of closed-form expressions to determine all of the subsets of the combined model which when fixed induce a null equi-output transformation group on the complement parameter sets.

3. Incorporation of an an accurate lens model such as (3.4).

4. replacement of stereo-cameras by tracking high precision laser interferometry (Prenninger et al., 1993) (Heeren and Veldpaus, 1992) (Newman and Osborn, 1993).

10.3 Canonical Solution Groups of Dextrous Manipulators

The kinematic mapping of a rigid open-link manipulator is a homomorphism between Lie groups. The homomorphism has solution groups that act on an inverse kinematic solution element. A canonical representation of solution group operators that act on a solution element of three and seven degree-of-freedom (*dof*) dextrous manipulators was determined by geometric analysis. Seven canonical solution groups were determined for the Cartesian yaw orientable seven *dof* manipulators and five solution groups for the Robotics Research K-1207 and Hollerbach arms. The solution element of a dextrous manipulator is a collection of trivial fibre bundles with solution fibres homotopic to the Torus. If fibre solutions are parameterised by a scalar, a direct inverse function that maps the scalar and Cartesian base space coordinates to solution element fibre coordinates may be defined. Such direct inverse parameterisation of a solution element may be approximated by a local linear map generated by an inverse augmented Jacobian correction of a linear interpolation. The action of canonical solution group operators on a local linear approximation of the solution element of inverse kinematics of dextrous manipulators generates cyclic and parallelisable solutions. Such an inverse kinematic solution representation also has approximation scaling properties that recommend it as a model of kinematic transformations in primate nervous systems.

Chapter 8 is the first work to decompose redundant manipulator inverse solutions into canonical solution groups that act an a local linear approximation of a minimal non-decomposable solution element. A lattice expansion of analytically determined inverse augmented Jacobians was combined with local linear interpolation to approximate the solution element and an error analysis of the approximation was also derived. A simple comparison with a nonparametric function approximation in §9.4 reveals that if b is the discretisation level of sampling of $\mathbb{R}$ and S, the total sampling requirements for the solution group solution element are order b^2 for the local linear approximation vs a nonparametric function approximation of the complete mapping sampling requirements of order b^7. The proposed solution may be several million times more storage and parameter determination complexity effecient than a nonparametric approximation.

The proposed solution may be competitive with state of the art solutions for dextrous manipulator inverse kinematics such as (Manocha and Canny, 1994) because it is cyclical, all solution branches are parameterised and it could be implementable as a RAM cache addressing solution. Verification

and validation of the proposed solution for the RR-K1207 is a non-trivial future research extension. Implementation of the solution for the RR-K1207 would require some attention to integration of a fixed Euler angle representation of orientation and a spherical coordinatisation of $\mathbb{R}^3$ with the with more usual coordinatisations of SE(3) embedded in manipulator control systems.

A modified (Ritter et al., 1992) local linear approximation of a dextrous arm homomorphism solution element combined with the canonical solution group operators defined in Chapter 8 was proposed in §9.4 as a model of inverse kinematics in nervous systems. It is hypothesised that solution group operators are represented in the posterior parietal cortex and the local linear approximation of the solution element is represented in the posterior parietal cortex or the cerebellum. The canonical solution groups in SE(3) are canonically represented in a fixed angle representation of SO(3) and a spherical coordinate representation of $\mathbb{R}^3$ attached to the shoulder. Hence this implies that this coordinate system is the shoulder representation system at least if not the head centred coordinate system. A future extension of this work could test this hypothesis by generation of an appropriate psychophysical and sensorimotor task experiment.

10.4 The Role of Nonparametric Regression in Robotics and Automation Systems

§5.5 described the optimal fitting of a radial basis function to a stereo-camera projection function which incorporates all difficult to model effects such as non-orthogonal optical axis, image centre offsets, depth of field variation, radial wide angle lens nonlinearities, image skew, shear, rotation, and scaling, and auto-focus on the LED. The hierarchical k-fold cross validation procedure to determine the radial basis function low bias and low variance parameters described in §5.2.1 is a completely automatic procedure to determine optimal nonparametric model complexity and parameter values for regression to an unbiased estimate of an underlying regression function. The automatic procedure is suitable for deployment by an engineer or software developer that is unfamiliar with low-dimensional nonparametric regression theory as an automatic black box nonparametric regressor.

§5.4 concluded that there were no hard commercial applications of nonparametric regression at kinematic, including path planning levels of manipulator control and few applications at dynamic control levels of robotics and automation systems for two reasons:

1. nonparametric solutions are easy to implement but difficult to verify and validate without recourse to advanced statistical methods. See (Bishop, 1995),

2. nonparametric solutions are difficult to scale.

The curse of dimensionality and the bias-variance tradeoff in nonparametric estimation are the major barriers to increasing functionalities of nonparametric robotics. The curse of dimensionality has important implications for nonparametric approximation of inverse kinematics of 7 *dof* dextrous manipulators. The curse of nonparametric regression dimensionality implies that most high dimensional regression spaces must be factorised into products of lower dimensional spaces. All of the nonparametric regressions on manipulator kinematic functions analysed in §5.3 have serious scaling problems if mapping symmetry is not factored into a nonparametric approximation strategy.

It is the conclusion of this work that domain analysis of a function to be approximated by a nonparametric model is more important than details of algorithms for optimisation of nonparametric model parameters, especially since the theory of optimal nonlinear regression and nonparametric

pattern classification is well understood. See (Seber and Wild, 1989; Ripley, 1996). A good problem domain analysis often leads to a regression or function approximation specification that is easy to fit, for instance with a k-nearest neighbour approximator with linear interpolation.

Nonparametric models may have greater application potential at higher levels of robotics and automation systems when analytical modelling fails, usually close to economic and production scheduling levels. For non-critical applications, i.e. where there are no reliability, safety, availability, and robustness control constraints, economic circumstances may dictate that for instance an automatic nonparametric regression solution that can be quickly and adequately deployed by an engineer or software developer is an adequate solution given solution cost and delivery time constraints.

Thus the final prediction of this thesis is that advance of robotic system functionality must be based on rigorous analysis and system engineering. Learning algorithms operating on nonparametric models are not a substitute for problem analysis and solution synthesis.

10.5 Scalabable Robot System Engineering

A prototype distributed object environment for point and click image-based tele-command of manipulators and stereo-cameras that supports rapid prototyping of numerical experiments in distributed system control was specified and implemented in Chapter 6. The environment was validated by a hierarchical k-fold cross validated calibration to Cartesian space of a radial basis function regression correction of an affine stereo model. The function of the prototype environment was to test object system analysis and design concepts at the implementation level. A future extension of this work at the object oriented design and analysis methodology level is to refine and extend the proposed Functional Object Composition Hierarchy (FOCH) system analysis and design methodology. In particular, it is necessary to determine structuring principles for incorporation of meta-object design patterns for meta-control and meta-data and architectural reflection into the FOCH specification iteration. The incorporation of these architectural meta patterns occurs during the top down functional analysis since it is the goal of the functional analysis to generate stable functional meta-control and meta-data interfaces to the object encapsulation of functional hot spots generated by the rate of change of certain underlying technologies such as compilers and operating systems.

The FOCH method for iterative sequential hierarchical functional and object subsystem analysis, specification and design is designed for specification of large distributed robotics and automation systems. It is necessary to refine the method by testing it on specification and implementation of a small FOCH system before progressing to large systems analysis. FOCH also lacks an architectural language to describe the functional and object components and for instance a high level description of the object component hot spots. Generation of a FOCH architectural language is a future research avenue. A FOCH architecture language should cleanly express solutions to several system design objectives - design for dependability, design for scalability and design for extendability.

The implementation experience and the design of the design methodology provided a foundation for definitions of basic design and performance requirements for scalable virtual micro-kernels that broker inter-Java-virtual-machine remote method invocations between components of secure fault-tolerant manageable open distributed agile Total Quality Managed ISO9000+ conformant Just in Time manufacturing systems.

The scalability of automated robot intensive production systems is constrained by limitations of analysis of sensorimotor coordination and the limitations of distributed system integration technologies. Robotics and automation systems vendor develop application development toolkits with inter-

faces to STEP, MMS, CCE, Profibus, CORBA, DCE, DCOM, and Java standards (ITU/ISO/IEC, 1996; Bender, 1993; Siegel, 1996; Usländer, 1995b; Usländer, 1995a; Owen, 1993). It is difficult to maintain high performance with these protocol stack interfaces to distributed heterogeneous computing systems. System integrators require a scalable secure manageable distributed real-time virtual machine. A candidate virtual machine technological language for implementation of ODP is CORBA. CORBA supports horizontal systems integration architectures. The potential of CORBA for deployment on large systems integration applications is threatened by Java and Active X. Based on a comparative technical analysis, it is predicted that if the performance of Java compilers improves and a scalable RMI broker is developed that scales to support implementation of large distributed real-time systems, then Java may become the main systems integration core language for enterprise mission critical applications.

Appendix A

The ISO/ITU Open Distributed Processing Reference Model

The (ITU/ISO/IEC, 1996) International Standards Organisation (ISO) and the International Telecommunications Union (ITU) Open Distributed Processing (ODP) distributed software engineering reference model (ITU/ISO/IEC, 1996) partitions the description of a distributed system into five separate interlocking specification viewpoints, each representing a different system perspective:

1. the **enterprise** viewpoint specifies the process and role organisation of a business system,
2. the **information** viewpoint specifies the information content of a system,
3. the **computational** viewpoint specifies the logical behaviour of communicating objects within a system,
4. the **engineering** viewpoint specifies the patterns and mechanisms that support distribution of a system,
5. the **technology** viewpoint specifies the infrastructural components that are used to build a distributed system.

The ODP specification is a single specification with five different viewpoint languages containing some invariants, e.g. lexical, across viewpoint languages. The five viewpoint specifications are linked by relations between conformance points, key terms. Conformance points define a single system instead of five decoupled systems. Many of the viewpoint specification links are provided implicitly by notations based on name correspondence. However some of the key constraints need to be specified explicitly. In an ODP architecture, constraints are placed on the relations between terms in the viewpoint languages. This imposes limits on sets of possible specification mappings. Most of the constraints are between terms in the computational and engineering languages. Conformance points are defined to create consistent interpretations of system components supporting ODP functions if these are defined separately, for instance by a system architect, an information architect, an engineering architect and corresponding software implementation groups at the technology level.

The Enterprise Viewpoint

The enterprise specification of a distributed system describes the behaviour of the system, roughly equivalent to the Operational Reference Model level of DFA, i.e. user interaction pre-conditions and

expectations. System interaction with the environment, for example users of the system, is analysed along with a description of the business processes which the distributed system supports. Concerns at the enterprise specification level are to describe the role of the system in the business and how business policies determine the functional behaviour of the system.

The Information Viewpoint

The information viewpoint models the system as a collection of abstract information components. Each component is defined in terms of the data held within it, its relation to other components within the system and what constraints govern either internal data consistency or the relational specification of the components. Each information component defines a typed set of attributes which constitute its internal data. Relations between components express information links between components. The cardinality of each relation must be specified. The definition of the information level specification has several aspects:

1. specification of **dynamic** schemata. These are equivalent to information state interaction charts, e.g. similar to Harel chart notation. See (Booch, 1994).

2. specification of **invariant** schemata. These are boundary conditions on object operation, e.g. $x < 10$ always.

3. specification of **static** schemata. These are equivalent to object constants, e.g. at time t, $x = 10$, always.

It is necessary to consider at the information viewpoint specification level how to test schemata conformance at the implementation, i.e. the technology, level. It is possible for instance to use predicate member functions within objects to verify static and invariant schemata and define levels of schemata invariance priority so that before critical dynamic operation schemata are tested, object member predicate functions are invoked to test object state is consistent, i.e. to test conformance of object state to information specification static and invariant schemata.

The Computational Viewpoint

The computational level decomposes a system into objects performing individual functions and interacting at well-defined interfaces. The ODP computational viewpoint defines the following computing component activities:

1. object creation and destruction,

2. interface creation and destruction,

3. interface trading,

4. interface binding.

An ODP computational model defines a system as a collection of communicating information components. Each component acts either as a source or sink of information and can initiate or respond to communication requests. Each component satisfies a contractual interface definition, derived from the many interactions a given component participates in. A computational interface of a component has three parts; a signature, a behaviour and an environmental contract. Interactions between components in the computational viewpoint can be one of three types:

1. **operations** - similar to local or remote procedure calls (invocations). Data can be passed bi-directionally between a caller a callee.

2. **streams** - transmission of continuous sequences of data between interfaces of separated systems.

3. **signals** - atomic interactions in which data and control can only pass from a caller to a callee.

Heterogeneous component interaction sequences may be defined with mixed operation response, stream or signal response to operation, stream and signal commands between components. Each interface type has a different type of signature:

1. an operational signature defines the set of operations supported at the interface and whether the interface is a client or server. Each operation is defined in terms of its typed parameters and return types.

2. a stream signature defines the set of flows supported at the interface and whether the interface is a producer or consumer of such flows

3. a signal signature defines the set of signals and whether the interface initiates or responds to each signal.

ODP computational interfaces also support the notion of sub-types, i.e. object oriented inheritance. A computational component can support multiple interfaces. The behaviour of a distributed system must support a dynamic relation between interface and component. An interface can be deleted and the component may expose only a subset of its relation interfaces. While distribution issues enter the specification chain at the computational viewpoint level, specification of interaction mechanisms is delegated to the engineering and technological viewpoint levels.

The Engineering Viewpoint

The ODP engineering viewpoint describes the logical distribution of a system's components across processing elements. The engineering specification level receives specification inputs from the enterprise, information, and computational specification levels. The fundamental entities in the engineering viewpoint are objects and channels. Objects are:

1. basic engineering objects (corresponding to computational level components) and

2. infrastructure objects (e.g. protocol objects).

3. A channel corresponds to a binding or a binding object in the computational specification and consists of three types of object:

   (a) a stub which supports both the marshalling and un-marshalling of parameter data for given interactions,

   (b) a binder which maintains the link between the interacting objects in the face of potential failure, re-location or other configuration failures

   (c) a protocol object which manages the actual transmission of data according to some mutually agreed communication protocol.

A channel must be created before physically distributed objects can interact. The engineering level specification is careful to avoid committing to a specific processing element granularity at the technological level in its basic definitions of

1. cluster - a set of related co-located engineering objects,
2. capsule - a set of clusters with an associate cluster manager for each cluster, a capsule manager, and the parts of the channels which connect to their interfaces,
3. nucleus - an extended OS supporting ODP,
4. node - a computer system.

The engineering level specification thus defines the following specification hierarchy

$$\text{activities} \subset \text{engineering objects} \subset \text{clusters} \subset \text{capsule} \subset \text{nucleus object} \subset \text{node}$$

with inter-cluster communication occurring via channels. A node is usually a single processor but it can also be a SIMD or MIMD node if this has an OS extended to support ODP. The engineering viewpoint provides:

1. coordinative functions for capsules, clusters, and objects:
   (a) event notification,
   (b) component groups and replication,
   (c) component migration,
   (d) transactions.
2. loading of capsule and cluster components onto processing elements:
   (a) static load optimisation,
   (b) dynamic load optimisation.
3. component management functions:
   (a) create/delete component,
   (b) create/delete processing elements.

The specification of clusters, capsules, nodes, and interaction channels may generates a large number of interfaces, any of which may be selected as a conformance testing point. Various interfaces provide different types of conformance testing:

1. the interface between protocol objects is an inter-working conformance point where for instance OSI testing of communication behaviour is defined.
2. interfaces internal to a node represent boundaries between software modules are programmatic reference points for testing compatibility and portability conformance.
3. interfaces to basic engineering objects allow conformance testing of the system envelope or application validation.

The Technology Viewpoint

The technology viewpoint describes the implementation of the ODP system as a configuration of hardware and software components. The technology viewpoint provides a link between the set of viewpoint specifications and an actual implementation. The technology viewpoint links the viewpoint specification stack with real implementations by listing standards that can provide necessary basic operations in implemented basic languages, e.g. transport protocols and compilers. The technology language provides extra information need to implement and test selected standard components and communications mechanisms. The technology viewpoint language defines:

1. configuration of components/objects:
   (a) hardware description,
   (b) software description.
2. bottom-up constraints on ODP higher levels:
   (a) technology cost/availability,
   (b) quality of service and response times,
3. Implementation Standards, e.g. DCE, CORBA, Active X, Java.
4. Expression of Conformance - Implementation of eXtra Information for Testing (IXIT).

A technology viewpoint language has various structuring rules for specification of:

1. a classification of technology relevant to ODP systems,
2. a definition of interfaces between objects, for instance in an implementation standard interface definition language (IDL) supporting:
   (a) definition of configuration of components on processing nodes at the engineering level,
   (b) definition of IXIT for conformance testing,
   (c) definition atomic terms of higher ODP terms, i.e. grounding of BNF definitions.
3. is largely undefined by the ODP specification.

A technology languages has an important role in the conformance testing process. It grounds the upper specification conformance definitions in a language which compiles to the binary level.

Distribution Functions and Transparencies

The ODP reference model includes a collection of functions required to support the needs of the computational language (e.g. the trading function) and the engineering language (e.g. the relocator). Important function groups for critical real-time distributed highly available and reliable fault tolerant systems are:

1. management of engineering structures - nodes, capsules, clusters, objects.
2. coordination of nodes, capsules, clusters, and objects:
   (a) checkpoint and recovery,
   (b) de-activation and reactivation,

(c) event notification,

(d) groups and replication,

(e) object migration,

(f) transactions, a superset of two phase non-blocking ACID commit.

(g) recoverability,

(h) permanence,

(i) performance.

3. a type repository, including interface type trader concept.

A principle goal of the ODP specification is to mask engineering and technology issues from system architects operating at the computational and informational levels, i.e. to provide transparent distribution. Levels of distribution transparency defined in ODP reference model are:

1. access transparency. This provides the same programming model in terms of data representation and invocation style for interactions between either local or remote objects.

2. failure transparency. This supports the standard reporting of failures within a system and well defined recovery mechanisms.

3. location transparency. This allows for two objects, a caller and callee to be bound together to achieve some interaction irrespective of the individual locations of the objects involved.

4. migration transparency which masks from an object the fact that the system has changed the object's location. The location of an object is transparent to the object itself. This can be difficult to implement if an object accesses peripherals since perhipheral access consistency must be maintained after object migration.

5. relocation transparency. This allows an interface to change its location without affecting all other objects to which it is bound. This is also useful if a connection is broken. A relocator service can be used to reestablish the connection.

6. replication transparency. This allows for a group of objects to support some functionality such that the group acts as a singleton object. Every object in the group has common state and all state updates preserve total update order.

7. persistence transparency. This allows for the activation and de-activation of objects from a process address space transparent to other objects in the system.

8. transaction transparency. This allows different interactions across a configuration of objects to be coordinated to achieve durable inter-object invocation integrity and consistency.

Since ODP is a comprehensive standardised approach to specification of distributed system functionalities, it may be suitable for combination with TQM, ISO9000+ and JIT specifications to generate integrated scalable Total Quality Managed ISO9000+ conformant open distributed agile Just in Time manufacturing systems.

Appendix B

A Mapping of Open Distributed Processing to CORBA

The Common Object Request Broker Architecture (CORBA) is a middleware software layer that facilitates inter-operation of objects, written in different languages, executing on different operating systems. See (Siegel, 1996; Mowbray and Zahavi, 1995). CORBA supports semi-transparent distributed pointer references to remote objects over heterogeneous transport protocols and operating systems. CORBA is standardised by the Object Management Group (OMG). The OMG has 700 computer application vendors and enterprise level consumer memebers. Iona Technologies, Sun Microsystems, IBM, Hewlett Packhard and Digital are the most important CORBA product vendor members of the OMG.

The CORBA 2.0 specification in principle can function as an ODP technology language. In this Appendix, enterprise, information, computational, engineering, and technology viewpoint concepts of ODP are mapped to CORBA concepts after a brief overview of basic CORBA concepts.

The CORBA Interface Definition Language, Core, and Basic Object Adapter

CORBA provides a remote object invocation infrastructure that abstracts over the revision details and variability between operating systems and inter-process communication mechanisms. The CORBA 2.0 specification has several basic elements:

1. an interface definition language (IDL),
2. bindings of IDL to object languages, e.g. C++, Java, Smalltalk,
3. an object invocation request transport core,
4. a basic object adapter, i.e. distributed object pointer de-referencing, and binding to 2.,
5. object system services, i.e. generic application support infrastructure,
6. vertical application domain facilities - i.e. application specific programming frameworks.

The CORBA specification is constructed around a metaphor of a 'software bus' with:

1. a distributed object invocation request transport core or remote invocation 'bus'.

2. system services and vertical facilities plugged into the core via basic object adapters, whose IDL descriptions are in turn mapped to different target language bindings, e.g. C++, Ada and Smalltalk.

See Figure B. CORBA object interfaces are described by a set of IDL typed operations or typed attributes. Attributes can be read-only or read-write. Operations are defined by name, and in terms of the return type, in parameters, out parameters and in-out parameters. IDL supports inheritance. Compiling an IDL interface generates an IDL stub for the client side and an IDL skeleton for the server. A stub supports a list of functions corresponding to the operations and attributes in the IDL interface. The functions of a stub are:

1. to encapsulate the marshalling of parameters for IDL invocations,
2. to compose the request buffer and transmit the data on the transport network,
3. to extract the return values from the reply buffer,
4. to relay the return values to the calling application.

On the server-side, an IDL compilation generates an IDL skeleton. The functions of the IDL skeleton are:

1. to extract the operation and incoming parameters from the request buffer,
2. to dispatch an incoming request to the target object addressed,
3. on completion of the invocation, to compose the reply buffer marshalling the output parameters and return values.
4. to return the marshalled results of the invocation.

Both stubs and skeletons are binder objects in ODP terminology. The IDL skeleton on the server-side is post-compiled to a target application binding language, e.g. C++, to create a Basic Object Adapter (BOA). A client-side stub partially supports location transparency since application code does need to distinguish between invocations on local objects and those on remote objects. The CORBA specification defines a scoping operator `CORBA::`, for remote invocations to distinguish them from local object de-referencing. This violates ODP transparency.

The CORBA 2.0 specification also defines a Dynamic Invocation Interface (DII), i.e. interpretation of IDL instead of compilation to stubs. A DII requires a corresponding Dynamic Skeleton Interface (DSI) on the server side to interpret the DII loaded request. The DII and DSI are useful in applications such as browsers where all target interfaces are not fully determined at compile time. A combination of the DII and the Interface Repository supports dynamic type determination of CORBA objects and run-time invocation composition, i.e. provides a foundation for complicated meta-data based distributed systems.

In addition to specification of IDL, Orb Core, static (stubs and skeletons) and dynamic (DSI and DII) binders, and Basic Object Adapter, the CORBA 2.0 specification defines several application object infrastructure services which are necessary to support ODP concepts. While there are sixteen CORBA 2.0 services defined, the following services provide CORBA with the concepts to operate as an ODP technology language - naming, events, lifecycle, persistence, transactions, concurrency, security and trader. The core services for ODP support are transactions, events, management and naming, initially deployed in a soft, i.e. high reliability and availability, real-time context. To contextualise the definition of CORBA support for ODP concepts, the most important of these services are briefly described.

The Naming and Trader Services

A naming service supports a global distributed object name reference space, independent of object location in a distributed system. Thus it supports location transparency. An object name has an associated IDL interface. Each object has a global object reference and can be registered with the naming service with a location independent hierarchical name. A process requiring access to an object, retrieves the object reference by name from the naming service. It then binds to the object and can invoke the operations and request the attributes of the IDL interface supported. The naming service supports binding to objects by name.

The Trader service is a superset of the naming service. The trader service defines an interface query language defined on federations of interfaces. A trader service for objects implementing a service to be bound by matching a set of query conditions. The trader service implements the ODP trader specification.

The Event Service

Binding between objects in ODP can also involve multiple objects. The default bind mechanism within CORBA supports is only 1-1 binding. Multiple object binding can be provided by the event service and also by the relationship service. The event service supports n-ary object binding for homogeneous objects supporting the same IDL interface. The event service supports a multicast model of communication. Event channels connect multiple event suppliers with multiple consumers. An event channel identifies a collection of event suppliers and consumers that are bound together. An event channel is a CORBA object. It can be named and registered with the naming service. Event consumers register with the event channel. Event suppliers register with the event channel. A process wishing to bind to an event group, retrieves the object reference of the event channel by name from the naming service and thereafter either responds to or initiates events from or to the other suppliers and consumers within the group.

The event service supports event based communication between multiple suppliers of events and their consumers. The event service presents a view of an event-based messaging environment between event suppliers and consumers who communicate via an event channel. Event suppliers and consumers can push or pull events, corresponding to whether they are asynchronously made aware of relevant events or whether they actively seek out events.

The Object Transaction Service

The object transaction service (OTS) provides support for ensuring that an invocation consisting of one or more operations on one or more objects satisfies the ACID transaction properties:

1. atomicity - all actions bound to a transaction are either all performed or none of them are performed.
2. consistency - all actions bound to a transaction are such that the effects of the operations are accurate, correct and valid with respect to the application and objects semantics.
3. isolation - all intermediate results of actions performed in a transaction are not visible outside of the transaction.
4. durability - if a transaction completes successfully, the results of its operations are permanent.

The OTS supports a two phase commit nested transaction transaction protocol. See (Bernstein et al., 1987). This is sufficient to implement most of the ODP transaction specification.

System Management

A flexible system management approach is required in an ODP system in order to manage:

1. system development and specification/implementation lifecycle,
2. application operation fault monitoring, diagnosis and recovery; system configuration accounting, performance and security. System operation management is defined on standardised protocols such as the Simple Network Management Protocol (SNMP), the Common Management Information Protocol (CMIP) or special purpose protocols.
   (a) applications development - distributed multi-compiler, multi-OS code generation,
   (b) network and infrastructure management, based on system management standards such as SNMP or CMIP.

   It is a complicated software engineering task to integrate the many structural and temporal levels of system management into a single management system. While there are several opposing technical approaches to CORBA system management standardisation, most of the approaches address some of the requirements for ODP system management.

CORBA Support of the ODP Computational Viewpoint

CORBA supports the basic component definitions at the ODP computational specification level:

(a) object creation and destruction - CORBA lifecycle and naming services,
(b) interface creation and destruction - CORBA interface repository, naming service and DII/DSI,
(c) interface trading - CORBA trader service,
(d) interface binding - CORBA bind.

Components in the ODP computational viewpoint have operational, signal and stream interfaces each supporting a different signature and sub-typing. These can be directly mapped to CORBA concepts:

(a) CORBA supports the operational interface by IDL operations, with interface inheritance,
(b) signal interfaces are represented by one way operations or by the event service,
(c) stream interfaces are being defined by the OMG:
   i. A CORBA operational interface can control non-CORBA streams, e.g over Asynchronous Transfer Mode (ATM) networks and Integrated Standard Digital Networks (ISDN).
   ii. the CORBA externalisation service also covers streaming, along the lines of C++ stream operator overloading.
   iii. in conjunction with the TINA (Telecommunications Information Network Architecture) definition of stream interfaces for audio/video streams,

An object in the computational viewpoint may have multiple interfaces and it can dynamically change its exported interfaces. This is supported in CORBA by dynamic binding and interface composition.

CORBA Support of the ODP Information Viewpoint

CORBA provides support for the information viewpoint level of components through their publicly viewable IDL data and indirectly through the bindings from IDL definitions to a target language such as C++. It is within the target binding implementation of CORBA objects that object states are completely defined and where internal data consistency checks may have to be carried out to satisfy ODP conformance criteria depending on the granularity of object information state specification. In most cases, publicly viewable component states, i.e. IDL windows onto target binding objects, is probably sufficient granularity. Relations between components are well expressed at the IDL level. Conformance testing at the technology viewpoint for CORBA can be done, either by assertion of pre and post conditions within the implementation of operations or by exposing those assertions as IDL operations. An invariant pre/post condition monitoring process within the distributed system can track condition violations. It is also possible for instance to use predicate member functions within objects to verify static and invariant schemata and define levels of schemata invariance priority so that before execution of critical dynamic operation schemata, object member predicate functions are invoked to test if object state is consistent, i.e. to test conformance of object state to specification static and invariant schemata.

CORBA Support of the ODP Engineering Viewpoint

Each of the concepts in the ODP engineering viewpoint can be mapped to appropriate concepts in CORBA:

(a) A node is mapped to a processor or multi-processor in the CORBA architecture.

(b) A nucleus is represented in a CORBA infrastructure as a combination of the operating system and its communication protocols and the core ORB library.

(c) A capsule is a group of objects which share an efficient communication mechanism, implemented within a CORBA environment as a server process with objects in the same address space, or perhaps a set of server processes communicating via shared memory.

(d) Within a capsule, logical groupings of objects, clusters, may be defined. In CORBA, such groupings are defined by the relationship service.

For the mechanisms of managing communication, a CORBA IDL stub represents a stub as defined in the engineering viewpoint. A binder is implemented by a combination of the naming service, which manages the logical mapping of name to object reference, and the ORB protocol which can support re-configuration in the case of object migration. CORBA services also support:

(a) coordinative functions for the above entities:

  i. event notification - CORBA event and notification service,

  ii. component groups and replication - currently undefined,

  iii. component migration - CORBA lifecycle service,

  iv. transactions - CORBA OTS.

(b) component management functions:

i. create/delete component - CORBA lifecycle service,

ii. create/delete processing elements - standard system and network management based on SNMP or CMIP.

CORBA services do not currently coherently support loading of components onto processing elements for dynamic load optimisation. Perhaps CORBA will eventually support dynamic performance optimisation.

CORBA Support of ODP Functions

There is a relatively tight conceptual correspondence between ODP system functions and CORBA services:

(a) Management can be mapped to CORBA management, e.g. based on Extended Common Management Functions (XCMF). CORBA management of objects servers server groups, ORB core, and hosts corresponds to ODP management of objects, clusters, capsule, nucleus, and nodes.

(b) Transactions - the CORBA OTS is transparent because of the use of the naming service to locate resource mangers controlling resources. CORBA 2 phase ACID commit is a subset of the ODP transaction abstraction.

(c) Type Repository - the CORBA naming service stores system type information. The scope of the naming service in the CORBA specification is not finalised. For instance, there is debate about the interaction of the naming service with typed vs un-typed event systems. If typed events are selected, there is discussion over whether it is appropriate to store typed event information in the distributed object de-reference naming repository or in a separate event naming repository.

(d) Trader - the CORBA trader provides a three level method for locating interfaces satisfying various query criteria, including transitive interface relations, in federations of naming services, i.e. partitioned object reference systems.

CORBA Support of ODP Conformance Testing

CORBA support for ODP conformance testing covers;

(a) Programmer defined pre and post conditions on object invocations. This is the programmer level of system exception coding.

(b) CORBA specification validation using a conformance testing language such as Assertion Definition Language (ADL). If a CORBA validation test suite is defined in ADL, ADL can also be extended to cover ODP conformance testing of pre and post conditions of corresponding technological, engineering, computational, and informational level invariances.

(c) The CORBA security service also defines valid object interactions as part of its authentication procedures. This provides a route for testing some authentication related aspects of conformance.

The CORBA object lifecycle specification also covers constraints on allowable operations on objects at various stages in their lifecycle.

CORBA Support of Distribution Transparencies

(a) Access transparency is supported by the client-side IDL stub. Invocations on remote objects use a similar programming model as an invocation on a local object except for the use of the `CORBA::` scope operator.

(b) Failure Transparency. As CORBA defines a set of exceptions for each of the potential failures in an invocation on a remote call, the client may have partial knowledge of the state of the invocation and well defined recovery mechanisms for each failure may be defined. A second invocation to an object of a de-activated server will re-activate the server. Few CORBA implementations robustly support failure transparency.

(c) Location transparency is supported by the naming service. The naming service maintains a mapping from global location-independent names to object references. A client can resolve a global name through the naming service to bind to the associated object reference independent of the object location. The location of that object is encapsulated in a proxy at the client, and is transparent to the application code.

(d) Migration Transparency. The data and behaviour of an object implementing an IDL interface is theoretically independent of its location. The object adapter in a server maintains the mapping from location to IDL implementation object. If an object is migrated, using the lifecycle service for example, the object adapter at the destination location will be updated to reflect the new object location. The actual implementation object remains unchanged. Few CORBA implementations robustly support migration transparency.

(e) Relocation transparency ensures that a client reference to an implementation object remains valid even after the implementation object has migrated to another location. In CORBA, this is achieved by a combination of the underlying protocol and the ORB implementation. The Internet InterORB Protocol (IIOP) has a message which returns the object reference to the object at the new location in response to an invocation on the object reference at the old location. In this way, a client will be updated to transparently refer to the new object at the new location. Few CORBA implementations robustly support relocation transparency.

(f) Replication Transparency. The current definition of CORBA does not support replication of servers transparently. Replication can be programmed at the application layer. There are sufficient mechanisms to group replica servers together and to maintain consistency across these servers. Replication is generally achieved by two protocols:

i. an underlying object group protocol such as ISIS (see e.g. (Mullender, 1989)).

ii. a multicast facility in the event service to manage groups.

Few CORBA implementations robustly support replication transparency.

(g) Persistence Transparency. CORBA allows for the definition of different object adapters for different types of object implementation activation and de-activation. A persistence object adapter can activate objects from persistent storage, and can also evict or de-activate these objects.

(h) Transaction Transparency. The Object Transaction service in CORBA supports transaction transparency. From an application viewpoint, all objects participating in a transaction must implement an IDL interface deriving from a Transactional Object interface. For all invocations to such transactional objects, a transaction context, identifying the current

transaction, is transparently attached to each invocation request buffer. This is encapsulated in the IDL stub and skeleton and the application code is not exposed to the transaction or its state.

CORBA has some specification design flaws, especially at the service interaction level, which may result in inefficient heavy vendor implementations that are unsuitable for mission critical enterprise computing.

Advantages and Disadvantages of the CORBA Technology Langauge

The advantages of CORBA systems are that:

1. IDL and horizontal IDL based system architectures simplify system specification, integration and maintenance,
2. IDL encourages seperation of programming concerns,
3. IDL and CORBA coverage of different operating systems and target binding languages makes it useful for integration of legacy applications after reverse and reengineering,
4. CORBA compilers are relatively cheap,
5. CORBA compilers more effecient than a custom distributed object solution.

The disadvantages of CORBA system are that:

1. CORBA systems are not completely transparent,
2. there are some specification lacunae and incoherency at the service interaction level,
3. specification by committees is often port pragmatic oriented rather than high performance oriented,
4. application programming interface is not minimal, especially compared to Java,
5. perhaps fundamentally unscalable.

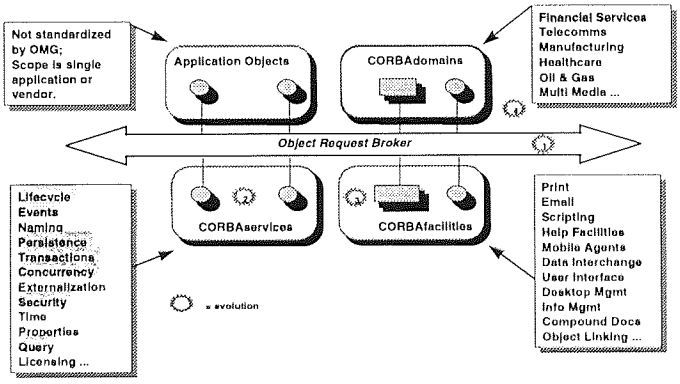

Figure B.1: The CORBA Reference Model.

Appendix C

Components of the Distributed Object Environment

Element	Function	Description
1	Application Scope	Prototype a distributed object robotic control system
1.1	Objectives	Math prototyping → hardware
1.2	Physical Realization	
1.2.1	Robot	Oxford Intelligent Machines RT100
1.2.2	Intrinsic Sensors	Optical joint position encoders
1.2.3	Cameras,Vision Pre-processing and Light	2 NTSC Cameras, SunVideo, LED
1.2.4	Dynamic Inversion	PID
1.2.5	Environment and Mock-ups	Blinds, overhead fluorescent strip
1.2.5.1	Robot Environment Layout	Obstacle free
1.2.6	Control Computer	SPARC classic **zippy**
1.2.7	Simulator	`Matlab`
1.2.8	Communication	UNIX sockets, `\dev\ttya`
2	Hardware Architecture	
2.1	Robot Controller Architecture	IP0 + IP1 boards
2.1.2	RT100 Driver	`\dev\ttya`
2.2	Sun-Video Driver	`\dev\rtvc`
2.3	Communication medium	Ethernet
3	Software System Architecture	
3.1	Basic Software	XIL, Xlib, Devguide, C++, `Matlab`
3.2	Process Interaction	Loosely-coupled synchronous
3.3	Inter-process Communication	sockets
3.4	Subsystems and Modules	
3.4.1	Robot Controller `robot++`	C/C++
3.4.2	Vision Pre-processing `image++`	C/C++
3.4.3	GUI Design `gui++`	XIL/Xlib/Devguide/C++
3.4.4	Process Scheduler	Shell script: sequential protocol
3.5	Interfaces	
3.5.1	I/F Scheduler	Parent-child synchronous
3.5.1.2	Exchange Data Formats	message and data class
3.5.2	`Matlab` I/F `robot++`	
v3.5.2.1	Exchanged Data	$\boldsymbol{\theta}_i$, $\mathbf{x}'_i$, status
3.5.2.2	Protocol	sequential blocking
3.5.3	`Matlab` I/F `image++`	
3.5.3.1	Exchanged Data	Command grab image, return $\mathbf{x}'_i$
3.5.3.2	Protocol	sequential blocking
3.5.4	`image++` I/F `gui++`	
3.5.4.1	Exchanged Data	raw or compressed images
3.5.4.2	Protocol	sequential blocking.
3.5.5	`gui++` I/F `robot++`	
3.5.5.1	Exchanged Data	commands and state information
3.5.5.2	Protocol	sequential blocking.
4	Nonparametric Modules	$\mathbb{R}^4 \rightarrow \mathbb{R}^3$
4.1	Architecture Principle	`Matlab` → C → DSP
4.2	Camera Calibration to Cartesian space	Cross validated $\mathbf{S} + \mathcal{F}$

Appendix D

The Kinematics of Planar 2-R

The planar 2**R** in figure 2.3 with link lengths $\{l_1, l_2\}, l_2 < l_1$ can reach points in $\mathfrak{W} \subset \mathbb{R}^2$ by varying θ_1, θ_2. $0 \leq \theta_1, \theta_2 \leq 2\pi$. $\mathfrak{W}$ is

$$\{(x, y) \in \mathfrak{W} \subset \mathbb{R}^2 | l_1 - l_2 \leq \sqrt{x^2 + y^2} \leq l_1 + l_2\}. \tag{D.1}$$

The forward kinematics are

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2), \tag{D.2}$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2). \tag{D.3}$$

From Figure 2.3 and the *law of cosines*[1], the geometric inverse kinematic solution is

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi + \theta_2). \tag{D.4}$$

If $\sqrt{x^2 + y^2} \leq l_1 + l_2$, the **elbow-up** solution for θ_2 is

$$\theta_2 = \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right), \tag{D.5}$$

and the **elbow-down** solution is $-\theta_2$. To determine[2] θ_1, let $\theta_4 = \text{Atan2(y,x)}$ in Figure 2.3. By the *law of cosines*

$$\theta_3 = \arccos\left(\frac{x^2 + y^2 + l_1^2 + l_2^2}{2l_1\sqrt{x^2 + y^2}}\right), \tag{D.6}$$

$$\theta_1 = \theta_4 \pm \theta_3. \tag{D.7}$$

The same inverse solution may be found by algebraic solution. (D.2) and (D.3) squared, added and simplified is

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos(\theta_2). \tag{D.8}$$

Hence

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}. \tag{D.9}$$

[1] If a triangle's angles are labeled a, b and c, where angle a is opposite side A, and so on, then the *law of cosines* is that $A^2 = B^2 + C^2 - 2BC\cos(a)$.

[2] Atan2(y, x) computes $\tan^{-1}(\frac{y}{x})$, the geometric inverse solution. Atan2 uses the signs of both x and y to determine the quadrant of the solution angle. Atan2$(y, x) \rightarrow (0, 2\pi)$.

If $\sqrt{x^2+y^2} \leq l_1 + l_2$, then

$$\sin(\theta_2) = \pm\sqrt{1-\cos^2(\theta_2)}, \tag{D.10}$$

$$\theta_2 = \text{Atan2}(\sin(\theta_2), \cos(\theta_2)). \tag{D.11}$$

To solve (D.2) and (D.3) for θ_1, let

$$k_1 = l_1 + l_2\cos(\theta_2), \tag{D.12}$$

$$k_2 = l_2\sin(\theta_2). \tag{D.13}$$

If

$$\rho = +\sqrt{k_1^2 + k_2^2}, \tag{D.14}$$

$$\lambda = \text{Atan2}(k_2, k_1), \tag{D.15}$$

$$k_1 = \rho\cos(\lambda), k_2 = \rho\sin(\lambda), \tag{D.16}$$

then (D.2) and (D.3) can be rewritten as

$$\frac{x}{\rho} = \cos(\lambda)\cos(\theta_1) - \sin(\lambda)\sin(\theta_1), \tag{D.17}$$

$$\frac{y}{\rho} = \cos(\lambda)\sin(\theta_1) + \sin(\lambda)\cos(\theta_1). \tag{D.18}$$

Hence

$$\cos(\lambda+\theta_1) = \frac{x}{\rho}, \tag{D.19}$$

$$\sin(\lambda+\theta_1) = \frac{y}{\rho}, \tag{D.20}$$

$$\lambda + \theta_1 = \text{Atan2}\left(\tfrac{y}{\rho}, \tfrac{x}{\rho}\right) = \text{Atan2}(y, x). \tag{D.21}$$

$$\theta_1 = \begin{cases} \text{Atan2}(y,x) - \text{Atan2}(k_2, k_1), \\ \text{Atan2}(y,x) - \text{Atan2}(l_2\sin(\theta_2), l_1 + l_2\cos(\theta_2)). \end{cases} \tag{D.22}$$

The determinant of planar 2-**R** Jacobian is

$$|\mathbf{J}| = \begin{vmatrix} -l_1\sin(\theta_1) - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{vmatrix}, \tag{D.23}$$

$$= l_1 l_2 \sin(\theta_2). \tag{D.24}$$

At $\mathbb{CPS}$, $|\mathbf{J}| = 0 \iff \theta_2 = 0, \pi$.

Appendix E

The Kinematics of Planar 3-R

Planar 3-**R** with link lengths $\{l_1, l_2, l_3\}, l_1 > l_2 > l_3$ can reach points in $\mathfrak{W} \subset \mathbb{R}^2$ by varying $3\mathbf{R}$ angles $\theta_1, \theta_2, \theta_3$. $0 \leq \theta_1, \theta_2, \theta_3 < 2\pi$. If θ link coordinate systems are assigned as for planar 2-**R**, $\mathfrak{W}$ is

$$\{(x, y) \in \mathfrak{W} \subset \mathbb{R}^2 | l_1 - l_2 - l_3 \leq \sqrt{x^2 + y^2} \leq l_1 + l_2 + l_3\}. \tag{E.1}$$

The forward kinematics are

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3), \tag{E.2}$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3). \tag{E.3}$$

(E.2)(E.3) are two equations in three variables. If one variable is selected as an independent variable, say θ_3, (E.2) and (E.3) can be solved for the other two dependent variables θ_1, θ_2 in terms of θ_3. Two solution functions f_1, f_2 define a parameterisation of the solution fibre

$$\theta_2 = f_1(x, y, \theta_3), \tag{E.4}$$

$$\theta_1 = f_2(x, y, \theta_2, \theta_3). \tag{E.5}$$

f_1, f_2 can be derived algebraically by exploiting an Argand representation of $\mathfrak{W} \subset \mathbb{R}^2$. A complex number be represented by Cartesian coordinates of the Argand plane $z = x + iy$ or by polar coordinates of the Argand plane $z = Re^{i\phi}$ where $\phi = \tan^{-1} \frac{y}{x}$ and $R = \sqrt{x^2 + y^2}$. The identity $e^{i\phi} = \cos\phi + i \sin\phi$ maps the Cartesian representation to the polar representation. Planar 3-**R** kinematics defined in a polar representation of the Argand plane are

$$z = x + iy = Re^{i\phi} = l_1 e^{i\theta_1} + l_2 e^{i(\theta_1 + \theta_2)} + l_3 e^{i(\theta_1 + \theta_2 + \theta_3)}. \tag{E.6}$$

A useful identity of polar representation power algebar is

$$Re^{i(\phi - \theta_1)} = l_1 + l_2 e^{i\theta_2} + l_3 e^{i(\theta_2 + \theta_3)} \tag{E.7}$$

$$= l_1 + e^{i\theta_2}(l_2 + l_3 e^{i\theta_3}). \tag{E.8}$$

Let

$$\delta e^{i\psi} = l_2 + l_3 e^{i\theta_3}, \tag{E.9}$$

then

$$\delta^2 = l_2^2 + 2l_2l_3\cos\theta_3 + l_3^2, \tag{E.10}$$

$$\delta = \pm\sqrt{l_2^2 + 2l_2l_3\cos(\theta_3) + l_3^2}. \tag{E.11}$$

$$\delta\cos\psi = l_2 + l_3\cos\theta_3, \tag{E.12}$$

$$\psi = \cos^{-1}\left(\frac{l_2 + l_3\cos(\theta_3)}{\delta}\right). \tag{E.13}$$

Substituting (E.9) into (E.7) and computing the modulus

$$Re^{i(\phi-\theta_1)} = l_1 + \delta e^{i(\theta_2+\psi)}. \tag{E.14}$$

The modulus squared of z is the product of z and z conjugate

$$R^2 = (l_1 + \delta e^{i(\theta_2+\psi)})(l_1 + \delta e^{-i(\theta_2+\psi)}) \tag{E.15}$$

$$= l_1^2 + 2l_1\delta\cos(\theta_2+\psi) + \delta^2. \tag{E.16}$$

$$\implies \cos(\theta_2+\psi) = \frac{R^2 - l_1^2 - \delta^2}{2l_1\delta}. \tag{E.17}$$

Hence θ_2 is a function of θ_3 after substitution and expansion

$$\theta_2 = \cos^{-1}\left(\frac{R^2 - l_1^2 - \delta^2}{2l_1\delta}\right) - \psi. \tag{E.18}$$

Substituting (E.9) into (E.7) and computing ϕ

$$\tan(\phi - \theta_1) = \frac{\mathrm{Im}(l_1 + \delta e^{i(\theta_2+\psi)})}{\mathrm{Re}(l_1 + \delta e^{i(\theta_2+\psi)})}. \tag{E.19}$$

$$\implies \tan(\phi - \theta_1) = \frac{\delta\sin(\theta_2+\psi)}{l_1 + \delta\cos(\theta_2+\psi)} \tag{E.20}$$

$$\implies \phi - \theta_1 = \tan^{-1}\left(\frac{\delta\sin(\theta_2+\psi)}{l_1 + \delta\cos(\theta_2+\psi)}\right). \tag{E.21}$$

Hence

$$\theta_1 = \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{\delta\sin(\theta_2+\psi)}{l_1 + \delta\cos(\theta_2+\psi)}. \tag{E.22}$$

When

$$\frac{R^2 - l_1^2 - \delta^2}{2l_1\delta} > 1.0, \tag{E.23}$$

$\cos^{-1}$ is defined in the Argand plane, hence there is no real solution. If $\theta_3 > 0$ the value of ψ must be negative. The solution branch structure is unclear from the algebraic solution. Geometric analysis of planar 3-**R** reveals that the above solution is for elbow down. The elbow up solution is

$$\theta_2 = -\pi + \cos^{-1}\left(\frac{R^2 - l_1^2 - \delta^2}{2l_1\delta}\right) - \psi, \tag{E.24}$$

$$\theta_1 = \tan^{-1}\frac{y}{x} + \tan^{-1}\frac{\delta\sin(\theta_2+\psi)}{l_1 + \delta\cos(\theta_2+\psi)}. \tag{E.25}$$

If $\zeta = \theta_3$ parameterises the solution element, then this defines an additional task function $\theta_3 = \theta_3$ in addition to (E.2) and (E.3). The derivatives of these three functions with respect to θ ($\frac{d\theta_3}{d\theta_3} = 1, \frac{d\theta_3}{d\theta_2} = 0, \frac{d\theta_3}{d\theta_1} = 0$) define $\mathbf{J}$ of the planar 3-$\mathbf{R}$ manipulator. $\mathbf{J}$ defined in a Cartesian coordinate system is defined by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) & -l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) & -l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) & l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) & l_3 \cos(\theta_1 + \theta_2 + \theta_3) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}. \tag{E.26}$$

Converting to polar coordinates $R = \sqrt{x^2 + y^2}$ and $\beth = \tan^{-1}(\frac{x}{y})$, the polar coordinate system augmented Jacobian is

$$\begin{bmatrix} \dot{R} \\ \dot{\beth} \\ \dot{\theta}_3 \end{bmatrix} = \mathbf{J}^{\Diamond} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}. \tag{E.27}$$

The analytical expansion of $\mathbf{J}^{\Diamond}$ represented as a vector formed by stacking the rows of $\mathbf{J}^{\Diamond}$ with top row first and bottom row last followed by a vector transpose operation is

$$\begin{bmatrix} \frac{2(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))(-l_1\sin(\theta_1)-l_2\sin(\theta_1+\theta_2)-l_3\sin(\theta_1+\theta_2+\theta_3))+2(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))}{2((l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2+(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2)^{\frac{1}{2}}} \\ \frac{2(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))(-l_2\sin(\theta_1+\theta_2)-l_3\sin(\theta_1+\theta_2+\theta_3))+2(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))(l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))}{2((l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2+(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2)^{\frac{1}{2}}} \\ \frac{2(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))-l_3\sin(\theta_1+\theta_2+\theta_3)+2(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))l_3\cos(\theta_1+\theta_2+\theta_3)}{2((l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2+(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2)^{\frac{1}{2}}} \\ \frac{\frac{-l_1\sin(\theta_1)-l_2\sin(\theta_1+\theta_2)-l_3\sin(\theta_1+\theta_2+\theta_3)}{l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3)} - \frac{l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3)}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))}{\frac{1+(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}} \\ \frac{\frac{-l_2\sin(\theta_1+\theta_2)-l_3\sin(\theta_1+\theta_2+\theta_3)}{l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3)} - \frac{l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3)}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}(l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))}{\frac{1+(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}} \\ \frac{\frac{-l_3\sin(\theta_1+\theta_2+\theta_3)}{l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3)} - \frac{l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3)}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}l_3\cos(\theta_1+\theta_2+\theta_3)}{\frac{1+(l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2)+l_3\cos(\theta_1+\theta_2+\theta_3))^2}{(l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2)+l_3\sin(\theta_1+\theta_2+\theta_3))^2}} \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}}. \tag{E.28}$$

$\mathbf{J}^{\Diamond^{-1}}$ is also simple to derive analytically but its expansion cannot be reproduced on A4.

Appendix F

Nonparametric Approximation of Planar 3-R Kinematics

Dimensionality reduction preprocessing of a function pre-image and image is a standard technique to reduce the curse of dimensionality (Bellman, 1961) of power-law sample scaling for uniform bound of error of nonparametric multi-variate function approximation. To recap, the curse of dimensionality is that in general the number of samples required for accurate nonparametric function approximation scales exponentially with the dimensionality d of the function pre-image. If each variable is discretised into M measurements, scaling for uniform bound of of regression error is of the order of M^d. It is standard for instance to do a principle components analysis of a nonparametric approximation of a regression function pre-image in order to reduce the bias and variance of regression. See (Bishop, 1995). A projection of an approximation data set pre-image onto the principle component directions of the data set is an inexact reduction of the dimensionality of a function approximation problem since information is lost in the projection. However the loss of information is compensated by a more accurate function approximation by a nonparametric model calibrated on a finite amount of data. See (Bishop, 1995).

Chapter 8 developed an exact dimensionality reduction of a mapping approximation problem by a solution group analysis of the mapping. In Chapter 8 the solution element was approximated analytically. The better representational effeciency of the solution group and analytically determined solution element over a full nonparametric approximation of a seven *dof* manipulator was quantified in §9.4 as:

1. One to ten million times more storage efficient.
2. At least several trillion times more computational complexity efficient since the complexity of determination of non-parametric regression parameters scales exponentially with dimensionality of the problem.

See also §5.3 and §5.4 for an analysis of the scaling limitations of nonparametric approximation of various aspects of inverse kinematics manipulators.

An analytical exact projection of an approximation data set of a mapping between Lie groups on to the canonical solution element manifold by the action of the solution group operators can also be approximated by a nonparametric model. For a given size data set, since an inexact dimensionality reduction can increase function approximation accuracy or reduce the bias and variance of a regression on a given size dataset, then by inference an exact dimensionality reduction also increases

approximation accuracy or reduces regression bias and variance on a given size dataset compared to a nonparametric approximation of the unprojected data set.

This Appendix closes the thesis with an analytical and numerical demonstration that this simple logical inference is correct. It is demonstrated that a nonparametric approximation of a particular data set projected to its canonical solution element manifold is two orders of magnitude more accurate than a nonparametric approximation of the unprojected data set. The data set is a sample of the inverse kinematics of a solution branch of planar 3-**R**.

Let $\mathcal{F}_e^{-1} : I_1 \subset \mathbb{R}^n \rightarrow R_1 \subset \mathbb{R}^m$ be an inverse kinematic solution branch e to be approximated. Let $\Upsilon_e^{-1} : I_2 \subset \mathbb{R}^o \rightarrow R_2 \subset \mathbb{R}^p$ be a solution element of $\mathcal{F}_e^{-1}$, such that $o < n$ and $p < m$. It is demonstrated that a nonparametric approximation of Υ_e^{-1} is up to two orders of magnitude more accurate over all of $\mathcal{F}_e^{-1}$ by the action of solution group operators than a complete nonparametric approximation of $\mathcal{F}_e^{-1}$ for the same approximation sample density over Υ_e^{-1} and $\mathcal{F}_e^{-1}$.

Let $\mathcal{F}_1$ be an approximation of $\mathcal{F}_e^{-1}$ and $\mathcal{F}_2$ be an approximation of Υ_e^{-1}. If $\mathcal{F}_1$ model complexity is equal to $\mathcal{F}_2$ model complexity and both $\mathcal{F}_1$ and $\mathcal{F}_2$ are optimally calibrated by the same nonparametric approximation parameter adjustment algorithm (see §5.1 to §5.2.2) to different data sets of the same sample density and distribution form over Υ_e^{-1} and $\mathcal{F}_e^{-1}$ respectively, the interpolation and extrapolation error $E(\mathcal{F}_1)$, e.g integrated L_2 error, of nonparametric approximation $\mathcal{F}_1$ of $\mathcal{F}_e^{-1}$ is a stochastic function[1] of sample density $P(I_1)$ over the approximation pre-image I_1, i.e. $E(\mathcal{F}_1) = f_1(P(I_1))$. Similarly for the approximation error of $\mathcal{F}_2$ over the solution element $E(\mathcal{F}_2) = f_2(P(I_1))$. To a first approximation, ignoring the interaction of the curvature of $\mathcal{F}_e^{-1}$ and the pre-image sampling probability distribution, by assuming a uniform distribution,

$$E(\mathcal{F}_1) \approx \frac{c}{P(I_1)}, \tag{F.1}$$

where c is a constant of proportionality. Similarly $E(\mathcal{F}_2) \approx \frac{c}{P(I_2)}$. The error of $\mathcal{F}_2$ evaluated over I_1, R_1 by the action of the solution group operators is approximately $E(\mathcal{F}_2)^{n-o+1}$.

Let $a < \infty$ be the number of $\mathcal{F}_1$ and $\mathcal{F}_2$ calibration data points. If a is sampled from a uniform regular or random pre-image sample distribution $p(I)$, a induces $P_a^1 = \frac{a}{\Lambda_1}$ sample density over the pre-image I_1 of $\mathcal{F}_e^{-1}$ where Λ_1 is the volume of I_1. Similarly a induces $P_a^2 = \frac{a}{\Lambda_2}$ sample density over the pre-image I_2 of the solution element Υ_e^{-1} of $\mathcal{F}_e^{-1}$ where Λ_2 is the volume of I_2. Since $o < n$, to a first approximation[2], by solution group factorisation $\Lambda_1 \approx \Lambda_2^{n-o+1}$. The sample density of $\mathcal{F}_1$ on the I_2 subset of I_1 is thus approximately $\frac{a}{\Lambda_2^{n-o+1}} < \frac{a}{\Lambda_2}$. Hence $P_a^1 < P_a^2$. Thus by (F.1), $E(\mathcal{F}_1)|_{I_2}$ evaluated on I_2 is greater than $E(\mathcal{F}_2)$ evaluated on I_2. This implies that the integrated error of the action of the solution group operators on $\mathcal{F}_2$ over I_1, R_1 is

$$E(\mathcal{F}_2)^{n-o+1} < (E(\mathcal{F}_1)|_{I_2})^{n-o+1} \approx E(\mathcal{F}_1). \tag{F.2}$$

Thus $\mathcal{F}_2$ with the solution group operators should be a more accurate approximation of $\mathcal{F}_e^{-1}$ than $\mathcal{F}_1$. (F.2) is a consequence of approximation (F.1). In the limit

$$a \rightarrow \infty, \frac{a}{\Lambda_2^{n-o+1}} - \frac{a}{\Lambda_2} \rightarrow 0, \text{ and } E(\mathcal{F}_2))^{n-o+1} - E(\mathcal{F}_1) \rightarrow 0, \tag{F.3}$$

but in practice nonparamtric models can only be calibrated to finite data.

A benchmark nonparametric approximation of a particular sub-homomorphism, the elbow up solution $\mathcal{F}_e^{-1}$ in $\mathfrak{W}_2$ of planar 3-**R** computed by the exact solution derived in Appendix E with

[1] See (Bishop, 1995; Seber and Wild, 1989) for a review of maximum likelihood and Bayesian methods for estimation of nonparametric regression estimation uncertainty, i.e. error bars, as a function of regression pre-image sample density.

[2] Exact for factorisation of hyper-cube subspaces of Euclidean manifolds but approximate for local factorisations of more irregular surfaces or for factorisation of Riemannian manifolds, such as $\mathbb{T}^n$.

$l_1 = 0.4m, l_2 = 0.3m, l_3 = 0.25m$, was undertaken to verify (F.2). Instead of using a function approximation method such as an interpolating polynomial or linear interpolation as in §8.1.3.2, a regression cross validation technique for minimisation of regression bias and variance was applied to a noiseless function approximation problem. Two universal function approximators, radial basis functions $\digamma^r$ and feed-forward networks $\digamma^f$, were benchmarked in a:

1. cross validated radial basis function approximation of:
   (a) $\digamma_1^r : (\theta_1^j, \theta_2^j) = \mathcal{F}_e^{-1}(x_j, y_j, \theta_3)$.
   (b) $\digamma_2^r : (\theta_1^i, \theta_2^i) = \Upsilon_e^{-1}(x_i, \theta_3)$ where the action of the solution group operators is determined by $x_i = \cos(-\eta)x_j - \sin(-\eta)y_j, \eta = \text{Atan2}(\frac{y_j}{x_j})$, and $\theta_1^i = \theta_1^j - \eta$. See §8.2.1.
   (c) cross validated feed-forward network approximation of
   (d) $\digamma_1^f : (\theta_1^j, \theta_2^j) = \mathcal{F}_e^{-1}(x_j, y_j, \theta_3)$.
   (e) $\digamma_2^f : (\theta_1^i, \theta_2^i) = \Upsilon_e^{-1}(x_i, \theta_3)$.

In the particular case of approximation of an inverse kinematic solution branch of planar 3-**R**, when $\digamma_1^r$ is compared with $\digamma_2^r$ or $\digamma_1^f$ is compared with $\digamma_2^f$, $n - o = 1$, $m - p = 0$. (F.2) implies that $E(\digamma_2^f)^2 < E(\digamma_1^f)$ and $E(\digamma_2^r)^2 < E(\digamma_1^r)$. (F.2) and (F.3) imply that for constant number of data points a, the integrated error of the $\digamma_1^f, \digamma_1^r$ approximations on the complete mapping will always be greater than the integrated error of the $\digamma_2^f, \digamma_2^r$ approximations respectively.

A cross validation of $\digamma^f, \digamma^r$ optimises the number of basis functions of $\digamma^r$ and $\digamma^f$ and the radial basis function variances $\boldsymbol{\sigma}^2$. A data set $\mathbb{D}$ containing 5000 data points was generated by a uniform distribution random sample of $\mathfrak{W}_2$ in a polar coordinate system and a random sample of θ_3 from $[0, 2\pi]$. Each data point d was determined by:

1. a radial distance $R_d = (2(l2 - l3)\xi) + l1 - l2 + l3$ where ξ is a uniform distribution random function with image $[0, 1]$.
2. a polar rotation $\mu_d = 2\pi\xi$.
3. $x_j^d = R_d \cos(\mu_d), y_j^d = R_d \sin(\mu_d)$.
4. $\theta_3^d = 2\pi\xi$.
5. $({}^d\theta_1^j, {}^d\theta_2^j) = \mathcal{F}_e^{-1}(x_j^d, y_j^d, \theta_3^d)$.
6. $({}^d\theta_1^i, {}^d\theta_2^i, x_i^d)$ are determined by solution group operators. See 1.(b) above.

$\mathbb{D}$ was bisectioned into a calibration data set $\mathbb{D}_c$ and a test data set $\mathbb{D}_t$. The error of $\digamma_1^f, \digamma_2^f, \digamma_1^r, \digamma_2^r$ on $\mathbb{D}_t$ is an approximation of the integrated errors $E(\digamma_2^r)^2, E(\digamma_1^r), E(\digamma_2^f)^2, E(\digamma_1^f)$ of the approximators on the function approximation problem. In order to assess the effect of the number of calibration data points and the random data variation on approximations $\digamma_1^r, \digamma_1^f, \digamma_2^r, \digamma_2^f$, $\mathbb{D}_c$ was partitioned into five pieces $\mathbb{D}_c^w, w = 1, \ldots, 5$ containing[3] 500 data points each. 10 subsets ${}^p\mathbb{D}_c^w$ of size $p = 50, 100, 150, \ldots, 500$ were selected from each $\mathbb{D}_c^w$. In order to cross validate $\digamma_1^r, \digamma_1^f, \digamma_2^r, \digamma_2^f$, each ${}^p\mathbb{D}_c^w$

[3]Initially a $\mathbb{D}_c$ of size 5,000 was used. This leads to a $\mathbb{D}_c^w$ of maximimum size of 1,000 data points. A singular value decomposition of the cross correlations between data set inputs and targets of $\digamma_2^r$ fails to converge over 550 data points even with very small radial basis function variances, of the order of 0.0001 of the average distance between basis function centres. At this level of sampling, the mean error on the $\digamma_2^r$ approximation is of the order of 0.01 radian and that of the $\digamma_1^r$ approximation is of the order of 1 radian. Two orders of magnitude error difference seemed sufficient to verify (F.2).

was partitioned into $m = 5$ equal calibration pieces ${}^{w}_{c}V^{p}_{i}$ and validation pieces ${}^{w}_{c}V^{p}_{-i} = {}^{p}\mathbb{D}^{w}_{c} \setminus {}^{w}_{c}V^{p}_{i}, i = 1, \ldots, 5$. Thus a total of 250 calibration data subsets and 250 validation subsets of $\mathbb{D}_c$ were generated.

To determine a search range for optimal H^*, the number of basis functions of the radial basis functions and feed-forward networks, a nonparametric regression heuristic is that the number of samples divided by the number of parameters per basis function leads to adequate nonparametric regression. To approximate (F^r_1, F^f_1) requires $5H$ parameters and to approximate (F^r_2, F^f_2) requires $4H$ parameters. The maximum number of calibration samples, 500, of output dimension two is effectively 1000 samples. The heuristic for estimation of basis function cardinality implies that the basis function upper bound of F^r_1, F^f_1 is 200 and of F^r_2, F^f_2 is 250 for a datset of size 500. The basis function search was defined on an inverval of the size of a given dataset divided by $5H$ ($4H$ respectively). This interval was divided into 5 to generate a search range for H^*.

Radial basis functions were initialised randomly on the data and feed-forward network basis functions were initalised randomly close to the origin. A single σ^2 spherical variance was used for all radial basis functions with a radius parameter. The search range for the radius parameter of σ^2 was defined by a sequence of ratios of the average random centre distance $\bar{d}^2$ to its nearest neighbour centre, $0.005\bar{d}^2, 0.01\bar{d}^2, 0.5\bar{d}^2, \bar{d}^2, 5\bar{d}^2$.

The feed-forward basis function parameters were calibrated using scaled congugate gradient optimisation. See (Press et al., 1992). The 100 optimal radial basis function models and the 100 optimal feed-forward networks were selected from a total set of 62,600 radial basis function models and 12,600 feed-forward network approximators and the errors on the optimal models on the the test data set were evaluated by the following algorithm:

for $p = 50, 100, \ldots, 500$ (size of calibration data set)
  for $w = 1, \ldots, 5$ (Calibration data partition $\mathbb{D}^w_c$)
    for $a = a^1, \ldots, a^{10}$ (the number of basis functions $H = f(\mathbb{D}^w_C)$)
      for $b = \bar{d}^2$ values of $005.2\bar{d}^2, 0.01\bar{d}^2, 0.5\bar{d}^2, \bar{d}^2, 5\bar{d}^2$ ($b = 1$ for F^f_1, F^f_2)
        for $i = 1, \ldots 5$ (for ${}^{w}_{c}V^{p}_{-i}$)
          for $l = 1, \ldots r = 5$ (for randomisation of basis function centres)
            Randomly initialise basis function centres.
            Calibrate F^r_1 on ${}^{w}_{c}V^{p}_{-i}$.
            Evaluate the validation error ${}^{w,p,1,r}E^{a,b}_{i,l}(\sigma^2_b, H_a)$ on ${}^{w}_{c}V^{p}_{i}$.
            Calibrate F^f_1 on ${}^{w}_{c}V^{p}_{-i}$.
            Evaluate the validation error ${}^{w,p,1,f}E^{a,b}_{i,l}(\sigma^2_b, H_a)$ on ${}^{w}_{c}V^{p}_{i}$.
            Map $(x^d_j, y^d_j, {}^{d}\theta^j_1)$ of ${}^{w}_{c}V^{p}_{-i}$ to$({}^{d}\theta^i_1, {}^{d}\theta^i_2, x^d_i)$ to get ${}^{w}_{c}V^{p^*}_{-i}$ and similarly for ${}^{w}_{c}V^{p^*}_{i}$.
            Calibrate F^r_2 on ${}^{w}_{c}V^{p^*}_{-i}$.
            Evaluate the validation error ${}^{w,p,2,r}E^{a,b}_{i,l}(\sigma^2_b, H_a)$ on ${}^{w}_{c}V^{p^*}_{i}$.
            Calibrate F^f_2 on ${}^{w}_{c}V^{p^*}_{-i}$.
            Evaluate the validation error ${}^{w,p,2,f}E^{a,b}_{i,l}(\sigma^2_b, H_a)$ on ${}^{w}_{c}V^{p^*}_{i}$.
          end (for l)
        end (for i)
        Determine ${}^{w,p,1,r}\bar{E}^{a,b}(\sigma^2_b, H_a)$ over the $r \times i$ experiments.
        Determine ${}^{w,p,1,f}\bar{E}^{a,b}(\sigma^2_b, H_a)$ over the $r \times i$ experiments.
        Determine ${}^{w,p,2,r}\bar{E}^{a,b}(\sigma^2_b, H_a)$ over the $r \times i$ experiments.
        Determine ${}^{w,p,2,f}\bar{E}^{a,b}(\sigma^2_b, H_a)$ over the $r \times i$ experiments.
      end (for b)
    end (for a)

Determine ${}^{w,p,1,r}\bar{E}^{c,d}(\boldsymbol{\sigma}_d^2, H_c) \leq {}^{w,p,1,r}\bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ for all a, b.

for e = 1, ...5

Calibrate ${}^{w,p}_{e}\boldsymbol{F}_1^r$ with structure $\boldsymbol{\sigma}_d^2, H_c$ and different random centre placement on ${}^{p}\mathbb{D}_c^w$.

end (for e)

Determine the average error $\bar{E}^{w,p,t,1,r}$ of the ${}^{w,p}_{e}\boldsymbol{F}_1^r, e = 1, \ldots, 5$ on $\mathbb{D}_t$.

Determine ${}^{w,p,2,r}\bar{E}^{c,d}(\boldsymbol{\sigma}_d^2, H_c) \leq {}^{w,p,2,r}\bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ for all a, b.

for e = 1, ...5

Calibrate ${}^{w,p}_{e}\boldsymbol{F}_2^r$ with structure $\boldsymbol{\sigma}_d^2, H_c$ and different random centre placement on ${}^{p}\mathbb{D}_c^w$.

end (for e)

Determine the average error $\bar{E}^{w,p,t,2,r}$ of the ${}^{w,p}_{e}\boldsymbol{F}_2^r, e = 1, \ldots, 5$ on $\mathbb{D}_t$.

Determine ${}^{w,p,1,f}\bar{E}^{c,d}(\boldsymbol{\sigma}_d^2, H_c) \leq {}^{w,p,1,f}\bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ for all a, b.

for e = 1, ...5

Calibrate ${}^{w,p}_{e}\boldsymbol{F}_1^f$ with structure $\boldsymbol{\sigma}_d^2, H_c$ and different random centre placement on ${}^{p}\mathbb{D}_c^w$.

end (for e)

Determine the average error $\bar{E}^{w,p,t,1,f}$ of the ${}^{w,p}_{e}\boldsymbol{F}_1^r, e = 1, \ldots, 5$ on $\mathbb{D}_t$.

Determine ${}^{w,p,2,f}\bar{E}^{c,d}(\boldsymbol{\sigma}_d^2, H_c) \leq {}^{w,p,2,f}\bar{E}^{a,b}(\boldsymbol{\sigma}_b^2, H_a)$ for all a, b.

for e = 1, ...5

Calibrate ${}^{w,p}_{e}\boldsymbol{F}_2^f$ with structure $\boldsymbol{\sigma}_d^2, H_c$ and different random centre placement on ${}^{p}\mathbb{D}_c^w$.

end (for e)

Determine the average error $\bar{E}^{w,p,t,2,f}$ of the ${}^{w,p}_{e}\boldsymbol{F}_2^f, e = 1, \ldots, 5$ on $\mathbb{D}_t$.

end (for w)

Compute the average error difference $\bar{E}^{p,t,r} = \frac{1}{5}\sum_{w=1}^{5}(\bar{E}^{w,p,t,2,r} - \bar{E}^{w,p,t,1,r})^2$.

Compute the average error difference $\bar{E}^{p,t,f} = \frac{1}{5}\sum_{w=1}^{5}(\bar{E}^{w,p,t,2,f} - \bar{E}^{w,p,t,1,f})^2$.

end (for p)

The results of the benchmark are presented in Figure F.1 to Figure F.16 and in Table F.1. Figure F.1 and Figure F.2 show the calibration and test data in $\mathfrak{W}_2$.

In Figure F.3 to Figure F.7, it can be seen that in general the RMSE difference of $\boldsymbol{F}_1^r - \boldsymbol{F}_2^r$ has a bump structure. The general trend (except for $\mathbb{D}_c^2$ in Figure F.4) is for error difference to grows rapidly between 50 and 100 data points and then declines steadily until it plateaus around 450 to 500 basis functions. See Figure F.14. The initial growth of error difference between 50 and 100 data points is caused by the on average large increase in accuracy of $\boldsymbol{F}_2^r$ calibrated on 100 data points compared to $\boldsymbol{F}_2^r$ calibrated on 50 data points, relative to accuracy gains as the number of data points is increased above 100. Exploratory experiments to explain why this occured indicate that it is due to the sensitivity of the approximations to the search range of the variance radius parameter. A more fine grain search of the variance parameter removes this bump but leads to a large increase in computational complexity. In order to plot a monotonic decreasing average error difference curve, it is necessary to use a more advanced basis function optimisation method such as Expectation Maximisation ((Dempster et al., 1977)) of the radial basis function centres combined with a line search on the radial basis variance radius parameter. Instead of a search on a fixed range of variance radii, it is possible to initialise the search on radius value r, calibrate the radial basis functions, and test the error on a data set, generate $r + \delta r$, recalibrate the radial basis functions, and evaluate the error gradient on the validation set for δr. With a gradient, the validation error can be minimised by a line search. See (Press et al., 1992). Best of all is to do a complete nonlinear optimisation of the centres and the elliptical individual basis function variance parameters (see (Bishop, 1995)) but this is computationally intensive, and impractical within the above cross validation experiment. The experiment verifies (F.2) even with

relatively optimal variance parameters, i.e. optimal defined with respect to the variance parameter search range set.

Figure F.15 is a plot of the ratio of the RMSE error of $\mathcal{F}_r^1$ to that of $\mathcal{F}_r^2$ as a function of the number p of calibration points. Figure F.15 shows that there is a large increase in the accuracy of $\mathcal{F}_r^2$ compared to $\mathcal{F}_r^1$ moving from 50 to 100 data points, followed by a decrease from 100 to 150 and then a steady increase. At 500 data points. the mean error on the $\mathcal{F}_2^r$ approximation is of the order of 0.01 radian and that of the $\mathcal{F}_1^r$ approximation is of the order of 1 radian. Thus for a given size of dataset, the radial basis approximation of the solution element is on average 2 order of magnitude more accurate on the test set than the radial basis approximation of the complete mapping. See Table F.1 for additional statistics on the the difference in accuracy of the complete nonparametric approximation compared to nonparametric approximation of a solution element for an equivalent sampling density. Figure F.3 to Figure F.7, Figure F.14, Figure F.5, and Table F.1 verify (F.2).

Figure F.8 to Figure F.12, Figure F.13, and Figure F.16 verify (F.2) on the benchmark of the feed-forward networks. Calibration of 12,600 feed-forward networks in matlab requires of the order of one month on a serial processor workstation. Figure F.13 shows that average error difference decrease on average as a function of the number of basis functions. Figure F.16 shows an average trend increase. The lack of monontonic average error decrease and ratio error increase is a result of local minima that are dependent on parameter initialisation combined with the problem of over-fitting the calibration set due to calibration sensitivity to model complexity. The average error of $\mathcal{F}_1^f$ on the test set for 500 data points is approximately 0.8 radians and of $\mathcal{F}_2^f$ is approximately 0.1 radians. Thus the average error of $\mathcal{F}_1^f$ on the test set is lower than that of $\mathcal{F}_1^r$ while the average error of $\mathcal{F}_2^f$ is higher than that of $\mathcal{F}_2^r$. Both benchmarks illustrate that the elementary logic of (F.2) is correct.

Both benchmarks also illustrate the general thesis of this work that nonparametric approximation is not a substitute for domain specific problem analysis for selection of appropriate problem solution representations and tools.

	RMSE($\mathcal{F}_1^r$) − RMSE($\mathcal{F}_2^r$)				RMSE($\mathcal{F}_1^f$) − RMSE($\mathcal{F}_2^f$)			
p	max($\mathbb{D}_c^i$)	mean($\mathbb{D}_c^i$)	var($\mathbb{D}_c^i$)	k_i	max($\mathbb{D}_c^i$)	mean($\mathbb{D}_c^i$)	var($\mathbb{D}_c^i$)	k_i
50	.3636	1.9331	1.8580	12.2742	2.3108	1.4542	0.4048	4.4235
100	8.2450	3.3814	7.4336	49.2451	2.8047	1.8820	0.5837	5.5909
150	1.8256	1.5057	0.0643	35.0361	1.9187	1.2092	0.2657	6.0536
200	2.1553	1.5356	0.2530	55.9194	2.2491	1.4684	0.2198	11.7166
250	1.8052	1.4369	0.0677	51.3054	1.0949	0.8875	0.0373	7.2062
300	1.8523	1.4566	0.1066	64.6303	1.8173	1.1909	0.1598	11.4694
350	1.8496	1.2893	0.1138	69.2138	2.1277	1.1381	0.3258	11.4194
400	1.8776	1.3849	0.0850	78.8035	2.8643	1.4899	0.7717	15.8339
450	1.4765	1.1848	0.0322	71.2766	1.2150	0.8318	0.0519	8.0965
500	1.4600	1.1809	0.0325	91.8393	2.8496	1.2030	0.8626	13.7892

Table F.1: Statistics of the RMSE differences of $\mathcal{F}_1^r - \mathcal{F}_2^r, \mathcal{F}_1^f - \mathcal{F}_2^f$ calibrated on $\mathbb{D}_c^1, \mathbb{D}_c^2, \mathbb{D}_c^3, \mathbb{D}_c^4, \mathbb{D}_c^5$ in Radians as a Function of the Number p of Data Points.

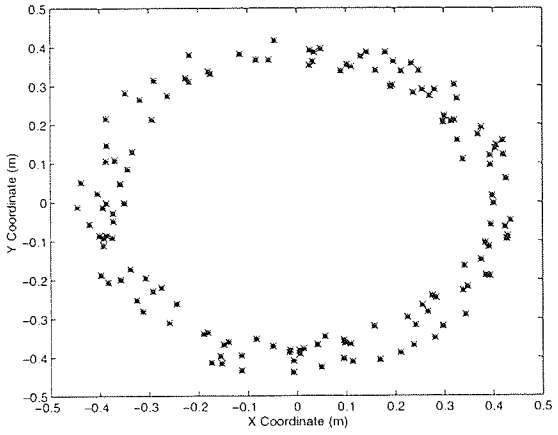

Figure F.1: A Plot of the Calibration Data in $\mathfrak{W}_2$.

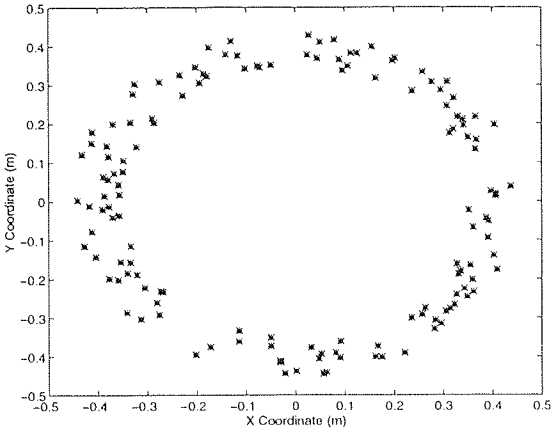

Figure F.2: A Plot of the Test Data in $\mathfrak{W}_2$.

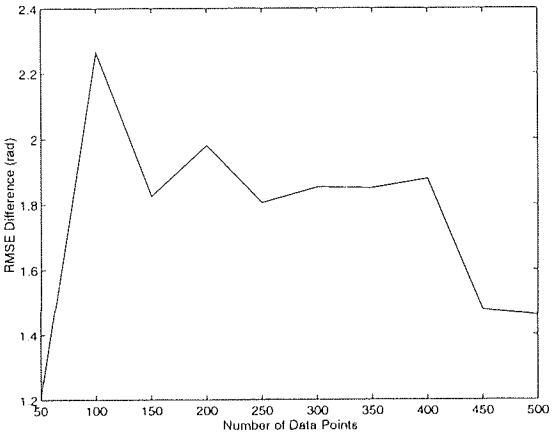

Figure F.3: A Plot of the RMSE of $\mathcal{F}_1^r$ minus the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^1$) on the Test Data as a Function of the Number p of Calibration Points.

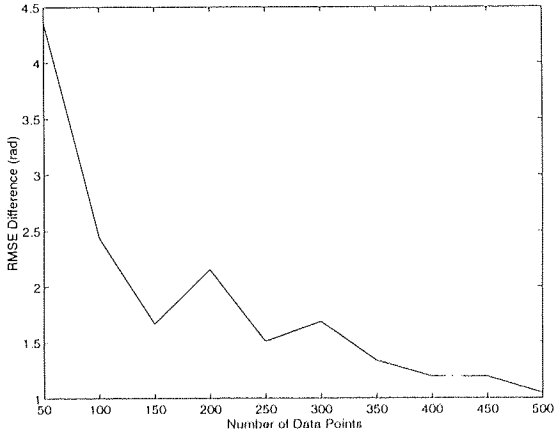

Figure F.4: A Plot of the RMSE of $\mathcal{F}_1^r$ minus the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^2$) on the Test Data as a Function of the Number p of Calibration Points.

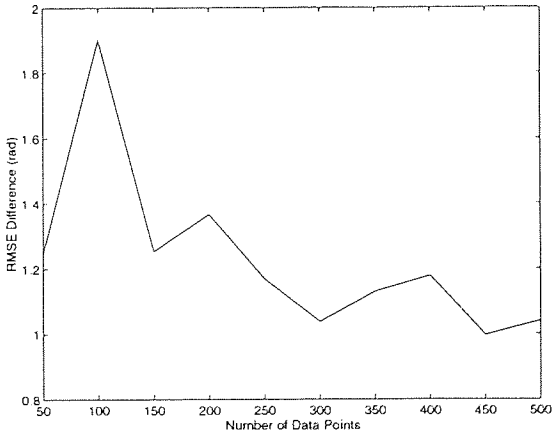

Figure F.5: A Plot of the RMSE of $\mathcal{F}_1^r$ minus the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^3$) on the Test Data as a Function of the Number p of Calibration Points.

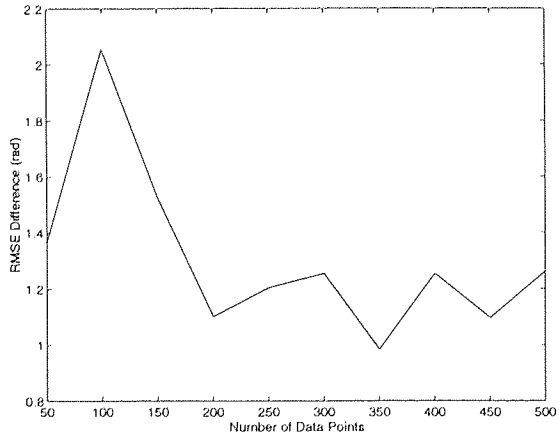

Figure F.6: A Plot of the RMSE of $\mathcal{F}_1^r$ minus the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^4$) on the Test Data as a Function of the Number p of Calibration Points.

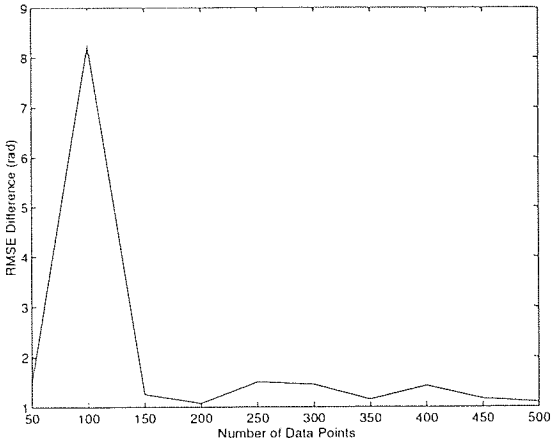

Figure F.7: A Plot of the RMSE of $\digamma_1^r$ minus the RMSE of $\digamma_2^r$ (for $\digamma_1^r, \digamma_2^r$ calibrated on $\mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Points.

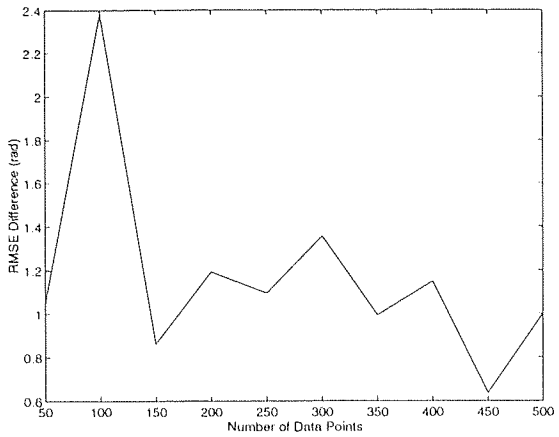

Figure F.8: A Plot of the RMSE of $\digamma_1^f$ minus the RMSE of $\digamma_2^f$ (for $\digamma_1^f, \digamma_2^f$ calibrated on $\mathbb{D}_c^1$) on the Test Data as a Function of the Number p of Calibration Points.

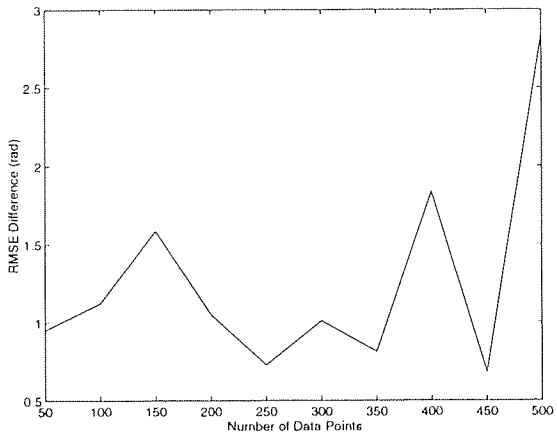

Figure F.9: A Plot of the RMSE of $\digamma_1^f$ minus the RMSE of $\digamma_2^f$ (for $\digamma_1^f, \digamma_2^f$ calibrated on $\mathbb{D}_c^2$) on the Test Data as a Function of the Number p of Calibration Points.

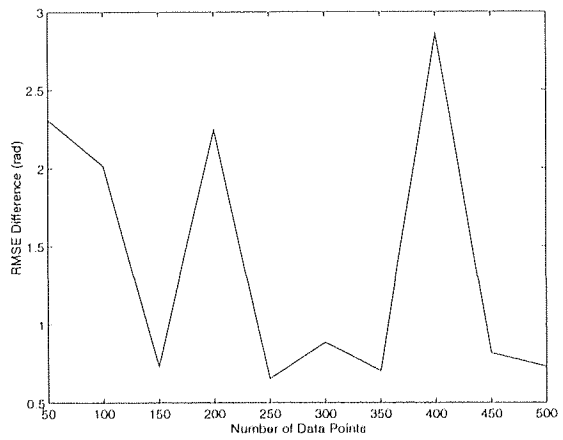

Figure F.10: A Plot of the RMSE of $\digamma_1^f$ minus the RMSE of $\digamma_2^f$ (for $\digamma_1^f, \digamma_2^f$ calibrated on $\mathbb{D}_c^3$) on the Test Data as a Function of the Number p of Calibration Points.

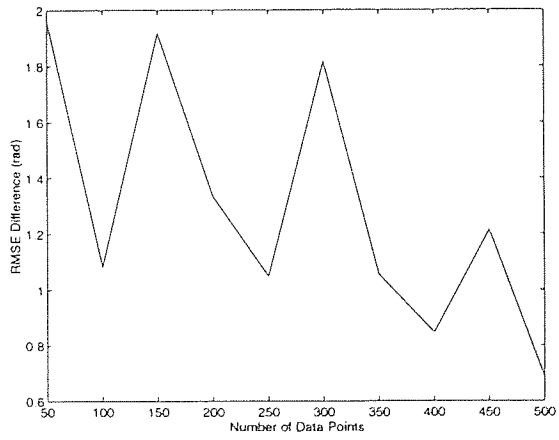

Figure F.11: A Plot of the RMSE of $\digamma_1^f$ minus the RMSE of $\digamma_2^f$ (for $\digamma_1^f, \digamma_2^f$ calibrated on $\mathbb{D}_c^4$) on the Test Data as a Function of the Number p of Calibration Points.

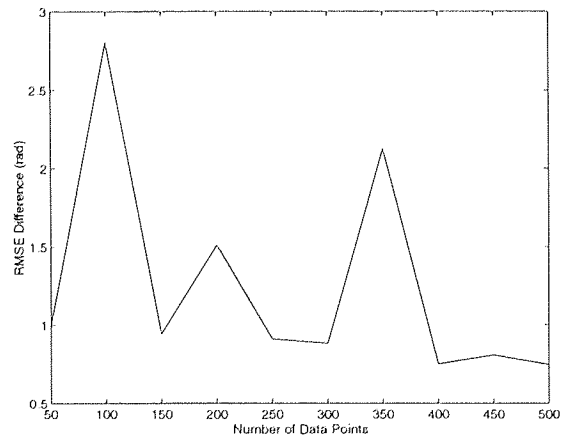

Figure F.12: A Plot of the RMSE of $\digamma_1^f$ minus the RMSE of $\digamma_2^f$ (for $\digamma_1^f, \digamma_2^f$ calibrated on $\mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Points.

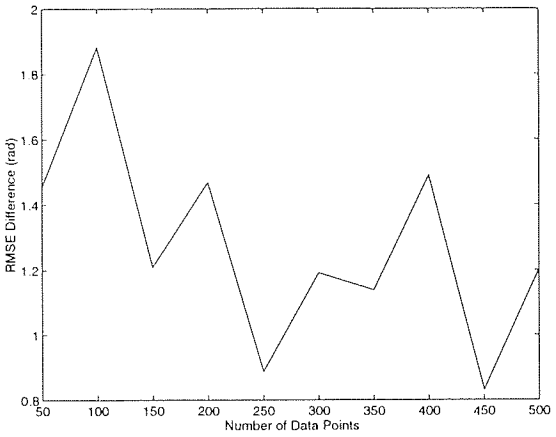

Figure F.13: A Plot of the average difference d_i of the RMSE $\mathcal{F}_1^f$ minus the RMSE of $\mathcal{F}_2^f$ (for $\mathcal{F}_1^f, \mathcal{F}_2^f$ calibrated on $\mathbb{D}_c^1, \mathbb{D}_c^2\mathbb{D}_c^3, \mathbb{D}_c^3, \mathbb{D}_c^4, \mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Points.

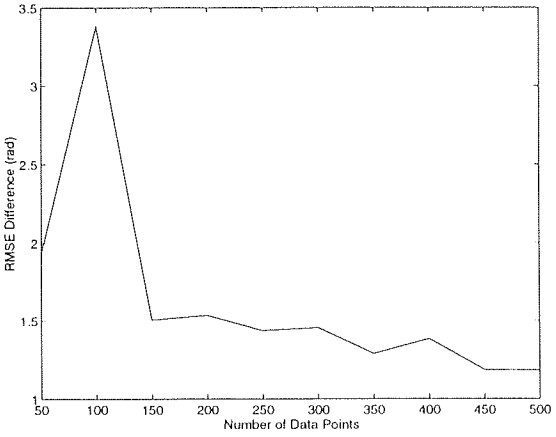

Figure F.14: A Plot of the average difference d_i of the RMSE $\mathcal{F}_1^r$ minus the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^1, \mathbb{D}_c^2\mathbb{D}_c^3, \mathbb{D}_c^3, \mathbb{D}_c^4, \mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Points.

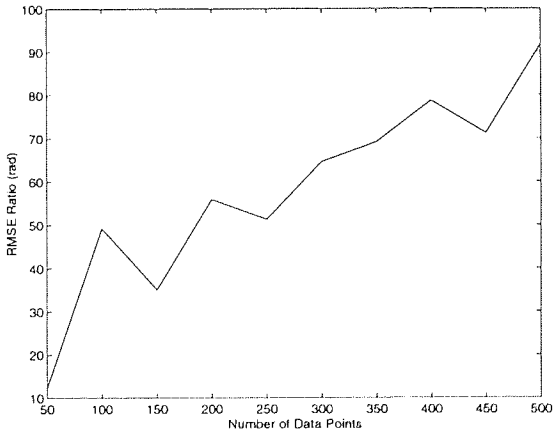

Figure F.15: A Plot of the ratio k_i of the average of the RMSE $\mathcal{F}_1^r$ to the average of the RMSE of $\mathcal{F}_2^r$ (for $\mathcal{F}_1^r, \mathcal{F}_2^r$ calibrated on $\mathbb{D}_c^1, \mathbb{D}_c^2\mathbb{D}_c^3, \mathbb{D}_c^3, \mathbb{D}_c^4, \mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Point.

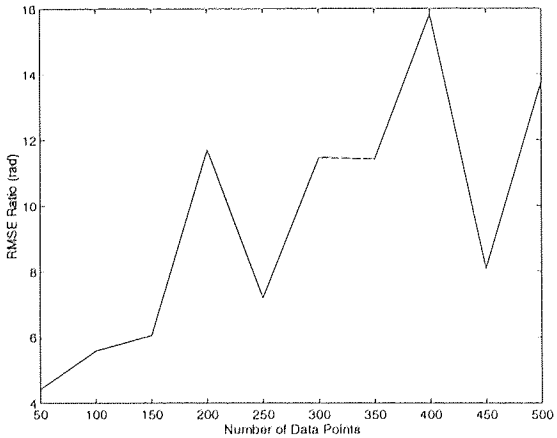

Figure F.16: A Plot of the ratio k_i of the average of the RMSE $\mathcal{F}_1^f$ to the average of the RMSE of $\mathcal{F}_2^f$ (for $\mathcal{F}_1^f, \mathcal{F}_2^f$ calibrated on $\mathbb{D}_c^1, \mathbb{D}_c^2\mathbb{D}_c^3, \mathbb{D}_c^3, \mathbb{D}_c^4, \mathbb{D}_c^5$) on the Test Data as a Function of the Number p of Calibration Points.

Bibliography

Albus, J. (1975). A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *American Society of Mechanical Engineers Journal of Dynamics, Systems, Measurement, and Control*, pages 220–227.

Albus, J., McCain, H., and Lumia, R. (1987). NASA/NBS Stanford reference model for telerobotic control system architecture (NASREM). Technical Report 1235, National Institute of Standards and Technology, Gaithersburg, MD.

Amador, E. (1995). A Robotics Research Environment. Master's thesis, Department of Computer Science and Applied Mathematics, Aston University.

Anderson, J. and Rosenfeld, E., editors (1988). *Neurocomputing: Foundations of Research*. MIT Press, Cambridge.

Apostol, T. M. (1967). *Calculus*, volume 1. Wiley, USA, second edition.

Arbib, M. (1987). Levels of modelling of mechanisms of visually guided behaviour. *Behavioural and Brain Sciences*, 10(407-465).

Arbib, M. (1995). *Handbook of Brain Theory and Neural Networks*. MIT Press, USA.

Asada, H. and Slotine, J.-J. (1986). *Robot Analysis and Control*. Wiley-Interscience, USA.

Baker, D. and Wampler, C. (1988). On the inverse kinematics of redundant manipulators. *International Journal of Robotics Research*, 7(4):3–21.

Baker, D. R. (1990). Some topological problems in robotics. *Mathematical Intelligencer*, 12(1):66–76.

Barhen, J., Gulati, S., and Zak, M. (1989). Neural learning of constrained nonlinear transformations. *IEEE Computer*, pages 67–76.

Barraquand, J. and Latombe, J. (1991). Robot motion planning: a distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649.

Basu, A. (1993). Active calibration. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 764–769.

Batchelor, R. (1994). *Henry Ford: Mass Production, Modernism and Design*. Manchester University Press, UK.

Begg, D. K., Fischer, S., and Dornbusch, R. (1994). *Economics*. McGraw-Hill, London, fourth edition.

Bekey, G. and Goldberg, K. (1991). *Neural Networks in Robotics*. Kluwer, MA.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour.* Princeton University Press, New Jersey.

Bender, K. (1993). *Profibus. The Fieldbus for Industrial Automation.* Prentice Hall International, USA.

Bennet, D. J., Geiger, D., and Hollerbach, J. M. (1991). Autonomous robot calibration for hand-eye coordination. *International Journal of Robotics Research*, 10(5).

Bernstein, P. A., Hadzilacos, V., and Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems.* Addison Wesley, USA.

Berthoz, A. (1991). Brain mechanisms in the perception and control of movement. In G.A. Orban, W. S. and Bernsen, N., editors, *Cognitive Neuroscience, Research Directions in Cognitive Science, European Perspectives*, volume 4, pages 43–71. Lawrence Earlbaum Associates, London.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods.* Prentice Hall, USA.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Booch, G. (1994). *Object Oriented Design with Applications.* Benjamin Cummings Publishing Company, Inc., USA.

Box, G., Hunter, W., and Hunter, J. (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building.* Wiley, New York.

Brady, M. (1985). Artificial intelligence and robotics. *Artificial Intelligence*, 26:79–121.

Break, R. and Haugen, O. (1993). *Engineering Real Time Systems. An Object Oriented Methodology using SDL.* Prentice Hall, Hemel Hempstead, UK.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, CA.

Burdick, J. (1988). *Kinematics and Design of Redundant Robot Manipulators.* PhD thesis, Department of Mechanical Engineering, Stanford University.

Caelli, T. (1985). *Visual Perception Theory and Practice.* Pergamon Press, Oxford.

Chaplin, J. and Krawiec, T. (1979). *Systems and Theories of Psychology.* Holt, Rinehart and Winston, New York, 4th edition.

Chen, A. M., Lu, H., and Hecht-Nielsen, R. (1993). On the geometry of feedforward neural network error surfaces. *Neural Computation*, 5(6):910–927.

Chen, Y. (1994). A new non-iterative approach for clustering. *Pattern Recognition*, 15:125–133.

Chomsky, N. (1954). Review of B.F. Skinner's verbal behaviour. *Language*, 35:26–58.

Cook, S. and Daniels, J. (1994). *Designing Object Systems - Object-Oriented Modelling with Syntropy.* Prentice Hall, USA.

Cooperstock, J. R. and Milios, E. E. (1993). An effeciently trainable neural network based vision-guided robot arm. *IEEE Computer*, 19:128–134.

Corke, P. I. (1994). Robotics toolbox for use with matlab. ftp ftp.mathworks.com.

Craig, J. (1989). *Introduction to Robotics.* Addison Wesley, USA.

Davis, M. (1958). *Computability and Unsolvability.* McGraw Hill, NY.

DeMers, D. (1993). *Learning to Invert Many-to-one Mappings.* PhD thesis, University of California, San Diego.

DeMers, D. and Kreutz-Delgado, K. (1991). Learning global topological properties of robot kinematic mappings for neural network-based configuration control. In Bekey, G. and Goldberg, K., editors, *Neural Networks in Robotics.* Kluwer, MA.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39:1–38.

Devijer, P. and Kittler, J. (1982). *Pattern Recognition: A Statistical Approach.* Prentice-Hall, Englewood Cliffs, NJ.

Doherty, C. (1993). Implementation of the Ritter and Schulten algorithm on transputers for control of a space robot. ESPRIT CONNY project deliverable 4.2. sub-contract from ERNO subsidiary of Messerschmidt-Blow-Blokum.

Dorf, R. C. (1988). *International Encyclopedia of Robotics, Applications and Automation.* Wiley, New York.

Dreyfus, H. (1979). *What Computers Can't Do.* Harper and Row, New York.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis.* Wiley, USA.

Eckmiller, R. (1989). Neural nets for sensorimotor trajectories. *IEEE Control Systems Magazine*, 9:53–59.

Edelman, G. (1989). *Neural Darwinism, the Theory of Neuronal Group Selection.* OUP, Oxford.

Elfving, A. and Kirchhoff, U. (1991). Design methodology for space automation and robotics systems. *European Space Agency Journal*, 15:149–164.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(13):313–325.

Everett, L. and Ives, T. (1993). A sensor used for measurements in the calibration of production robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 174–179.

Feldman, A. G. (1986). Once more on the equilibrium point hypothesis (λ model) for motor control. *Journal of Motor Behaviour*, 18:17–54.

Fikes, R. and Nilsson, N. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208.

Flanagan, D. (1996). *Java in a Nutshell.* O'Reilly and Associates, CA.

Foucault, M. (1980). *Power/Knowledge.* Pantheon, NY.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–141.

Fu, K., Gonzalez, R., and Lee, C. (1987). *Robotics, Control, Sensing, Vision and Intelligence*. McGraw-Hill, USA.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.

Gemignani, M. C. (1972). *Elementary Toplogy*. Dover, Toronto.

Gibson, J. J. (1968). *The Senses Considered as Perceptual Systems*. George Allen and Unwin, London.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston.

Ginsburg, H. and Opper, S. (1969). *Piaget's Theory of Intellectual Development*. Prentice Hall, NJ.

Guthrie, E. (1952). *The Psychology of Learning*. Harper and Row, NY.

Harrison, A. (1992). *Just-In-Time Manufacturing in Perspective*. Prentice-Hall, London.

Hashimoto, H., Kubota, T., Kudou, M., and Harashima, F. (1992). Self-organising visual servo system based on neural networks. *IEEE Control Systems Magazine*, 12:3–31.

Heeren, T. and Veldpaus, F. (1992). An optical system to measure the end effector position for on-line control purposes. *International Journal of Robotics Research*, 11(1):53–63.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Addison Wesley, USA.

Hilbert, D. and Cohn-Vossen, S. (1932). *Geometry and the Imagination*. Chelsea Publishing Company, NY.

Hilgard, E. R. and Bower, G. H. (1975). *Theories of Learning*. Prentice-Hall, Englewood Cliffs, NJ. fourth edition.

Hoffman, W. C. (1966). The Lie algebra of visual perception. *Journal of Mathematical Psychology*, 3:65–98.

Hollerback, J. (1990). Planning of arm movements. In Osherson, D., Kosslyn, S., and Hollerbach, J., editors, *An Invitation to Cognitive Science. Visual Cognition and Action*, volume 2, pages 183–211. MIT Press: Cambridge, MA.

Horn, B. (1986). *Robot Vision*. MIT Press, USA.

Hull, C. (1943). *Principles of Behaviour*. Appleton-Century-Crofts, NY.

Hutchinson, N. and Peterson, L. (1991). The x-kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17(1):64–76.

ITU/ISO/IEC, editor (1996). *Open Distributed Processing - Reference Model. International Telecommunications Union standard X.901...194. International Standards Organisation standard CD 10746-1...4.*

Jordan, M. and Rumelhart, D. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354.

Kandel, E. and Schwartz, J. (1985). *Principles of Neural Science*. Elsevier Science.

Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchial neural network model for control and learning of voluntary movement. *Biological Cybernetics*, pages 169–185.

Kelley, J. (1985). Principles of the functional and anatomical organisation of the nervous system. In Kandel, E. and Schwartz, J., editors, *Principles of Neural Science*, page 214. Elsevier Science, USA. Figure 19.3.

Kelly, G. (1955). *The Psychology of Personal Constructs*. Norton, NY.

Kircanski, M. V. (1993). Combined analytical pseudoinverse kinematic solution for simple redundant manipualtors and singularity avoidance. *International Journal of Robotics Research*, 12(2):188–196.

Knuth, D. (1973). *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison Wesley, USA.

Kohonen, T. (1989). *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3^{rd} edition.

Kreutz-Delgado, K., Long, M., and Seraji, H. (1992). Kinematic analysis of 7 *dof* manipulators. *International Journal of Robotics Research*, 11(5):469–481.

Kuperstein, M. and Rubinstein, J. (1989). Implementation of an adaptive neural controller for sensory-motor coordination. *IEEE Control Systems Magazine*, 9:25–34.

Latombe, J. (1991). *Robot Motion Planning*. Kluwer, MA.

Lee, C. G. (1991a). On the parallel algorithms for robotic computations. In Lee, C. G., editor, *Sensor-Based Robotics: Algorithms and Architectures*, pages 240–279. Springer Verlag, Heidelberg.

Lee, C. G., editor (1991b). *Sensor-Based Robots: Algorithms and Architectures*. Springer Verlag, Heidelberg.

Ljung, L. (1987). *System Identification. Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ.

Lowe, D. (1995). Radial basis function networks. In Arbib, M., editor, *The Handbook of Brain Theory and Neural Networks*, pages 134–151. MIT Press, Cambridge, MA.

Lyotard, J.-F. (1984). *The Postmodern Condition: A report on Knowledge*. Manchester University Press, Manchester.

Manocha, D. and Canny, J. (1994). Effecient inverse kinematics for general 6-**R** manipulators. *IEEE Transactions on Robotics and Automation*, 10(5):648–657.

Mansfield, E. (1989). The diffusion of industrial in Japan and the United States. *Research Policy*, 18:183–192.

Mardia, K., Kent, J., and Bibby, J. (1979). *Multi-variate Analysis*. Academic Press, San Diego. Nineth Printing 1994.

Marr, D. (1982). *Vision*. Freeman, San Francisco.

McCormick, E. J. (1981). *Industrial Psychology.* Allen and Unwin, London, seventh edition.

McCulloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5. Reprinted in (Anderson and Rosenfeld, 1988).

Mel, B. (1990). *Connectionist Motion Planning.* Academic Press, London.

Miller, W. (1989). Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Transactions on Systems, Man and Cybernetics*, 19(4):825–831.

Miller, W. T., Hewes, R. P., Glanz, F., and Kraft, L. (1990). Real-time dynamic control of an industrial manipulator using neural-network-based learning controller. *IEEE Transactions on Robotics and Automation.*

Miller III, W. T., Sutton, R. S., and Werbos, P. J. (1990). *Neural Networks for Control.* MIT Press, Cambridge: MA.

Mooring, B., Roth, Z., and Driels, M. (1991). *Fundamentals of Manipulator Calibration.* Wiley, New York.

Mowbray, T. J. and Zahavi, R., editors (1995). *The Essential CORBA. Systems Integration Using Distributed Objects.* John Wiley and Sons, Inc., USA.

Mullender, S. (1989). *Distributed Systems.* Addison Wesley, NY.

Naimark, M. and Stern, A. (1982). *Theory of Group Representations.* Springer Verlag, London.

Nakamura, Y. (1991). *Advanced Robotics: Redundancy and Optimisation.* Addison-Wesley, NY.

Newman, W. and Osborn, D. (1993). A new method for kinematic parameter calibration via laser line tracking. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 160–165.

Ng, M. (1995). An X Interface to Robot Sensor and Control Data. Master's thesis, Department of Computer Science and Applied Mathematics, Aston University.

Olver, P. (1993). *Applications of Lie Groups to Differential Equations.* Springer Verlag, London.

Olver, P. (1995). *Equivalence, Invariants, and Symmetry.* Cambridge University Press, USA.

Osserman, R. (1986). *A Survey of Minimal Surfaces.* Dover, USA.

Owen, J. (1993). *STEP - An Introduction.* Information Geometers, NY.

Paul, R. P. (1981). *Robot Manipulators: Mathematics, Programming and Control.* MIT Press: Cambridge MA.

Pavlov, I. (1927). *Conditioned Reflexes.* Clarendon Press, London.

Penfield, W. and Roberts, L. (1959). *Speech and Brain mechanism.* Princeton University Press, Princeton: NJ.

Penrose, R. (1989). *The Emperor's New Mind.* Oxford University Press, Oxford.

Piaget, J. (1972). *The Principles of Genetic Epistemology.* Routledge and K. Paul, London. Translated from the French by Wolfe Mays.

Pieper, D. (1968). *The Kinematics of Manipulators Under Computer Control.* PhD thesis, Stanford University.

Prenninger, J., Vincze, M., and Gander, H. (1993). Contactless position and orientation measurement of robot end-effectors. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 180–185.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipies in C.* Cambridge University Press, Cambridge, UK, second edition.

Putnam, H. (1981). *Reason, Truth and History.* Cambridge University Press, Cambridge, UK.

Putnam, H. (1987). *Putnam's Theorem: Chapter 15*, volume Women, Fire and Dangerous things: What Categories reveal about the mind. University of Chicago Press, Chicago.

Renders, J.-M., Rossibnol, E., Bequet, M., and Hanus, R. (1991). Kinematic calibration and geometrical identification for robots. *IEEE Transactions on Robotics and Automation*, 7(6):721–731.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, Trumpington Street, Cambridge.

Ritter, H., Martinetz, T., and Schulten, K. (1992). *Neural Computation and Self-Organising Maps.* Addison Wesley, USA.

Rizzolatti, G. (1991). Cortical control of movements: new research trends. In Orban, G. A., Singer, W., and Bernsen, N. O., editors, *Cognitive Neuroscience, European Research Perspectives: 4.* Lawrence Erlbaum, London.

Rohwer, R. J. (1994). Neural networks. In Michie, D., Spiegelhalter, D., and Taylor, C., editors, *Machine Learning, Neural and Statistical Classification.* Prentice-Hall, UK.

Rosenblatt, F. (1962). *Principles of Neurodynamics.* Spartan, New York.

Rueckl, J., Cave, K., and Kosslyn, S. (1990). Why are 'what' and 'where' processed by separate cortical visual systems? a computational investigation. *Journal of Cognitive Neuroscience*, 2(1):42–56.

Rumbaugh, J. (1991). *Object-Oriented Modeling and Design.* Prentice Hall, USA.

Rumelhart, D. and McClelland, J., editors (1986). *Parallel Distributed Processing*, volume 1 and 2. MIT Press, Cambridge, MA.

Samson, C., Borgne, M. L., and Espiau, B. (1991). *Robot Control. The Task Function Approach.* Oxford University Press, New York.

Samuel, A., McAree, P., and Hunt, K. (1991). Unifying screw geometry and matrix transforms. *International Journal of Robotics Research*, 10(5):454–472.

Sanner, R. and Slotine, J. E. (1995). Stable adaptive control of robot manipulators using neural networks. *Neural Computation*, 7(4):753–790.

Saylor, J. H. (1991). *Total Quality Management Field Manual.* McGraw-Hill, New York.

Schwartz, J. T. and Sharir, M. (1983). On the 'piano movers problem': II. general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4:51–96.

SDL (1993). *Specification and Description Language (SDL)*. International Technical Union, Geneva, Switzerland. Recommendation Z120.

Searle, J. (1981). Minds, brains and programs. *Behavioural and Brain Sciences*, 3(6):417–424.

Seber, G. and Wild, C. (1989). *Nonlinear Regression*. Wiley, London.

Shaw, M., Deline, R., Klein, D., Ross, T., Young, D., and Zelesnik, G. (1995). Abstractions for software architectures and tools to support them. *IEEE Transactions on Software Engineering*, 21:135–163.

Siegel, J., editor (1996). *CORBA Fundamentals and Programming*. John Wiley and Sons, Inc., USA.

Skinner, B. (1953). *Verbal Behaviour*. Appleton-Century-Crofts, NY.

Smith, B. (1991). The owl and the electric encyclopaedia. *Artificial Intelligence*, 47:251–288.

Sontag, E. (1990). *Mathematical Control Theory*. Springer Verlag, USA.

Stone, H. (1987). *Kinematic Modelling, Identification and Control of Robotic Manipulators*. Kluwer Academic, USA.

Tapper, C. (1989). *Computer Law*. Longman Group, Essex, UK.

Thorndike, E. (1949). *Selected Writings from a Connectionist's Psychology*. Appleton-Century-Crofts, NY.

Tikhonov, A. and Arsenin, V. (1977). *Solutions of Ill-Posed Problems*. Wiley, USA.

Tolman, E. (1932). *Purposive Behaviour in Animals and Men*. Appleton Century-Crofts, NY.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society 2nd Series*, 42:230–265.

Universal Machine Intelligence Ltd. (1987). *Inside RTX, Using Intelligent Peripherals Communicating, Programming RTX Using the Forth Library*. Universal Machine Intelligence Ltd., Oxford, UK, second edition.

Usländer, T. (1995a). *CCE: An integration Platform for Distributed Manufacturing Applications*. Springer-Verlag, Berlin.

Usländer, T. (1995b). *Manufacturing Message Specification: A Communication Language of Manufacturing*. Springer-Verlag, Berlin.

van der Smagt, P. (1994). *Visual Robot Arm Guidance Using Neural Networks*. PhD thesis, University of Amsterdam.

Varela, F., Thompson, E., and Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge: MA.

Vernon, D. (1991). *Machine Vision - Automated Visual Inspection and Robot Vision*. Prentice Hall, London.

Walter, J. and Schulten, K. (1993). Implementation of self-organising neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks*, 4(1):86–95.

Wampler, C. W. (1988). The inverse function approach to kinematic control of redundant manipulators. In *Proceedings of American Control Conference*, pages 817–823, Atlanta.

Warwick, K., Irwin, G., and Hunt, K., editors (1992). *Neural Nets for Control and Systems.* Peter Peregrinus Ltd., UK.

Weyl, H. (1946). *The Classic Groups.* Princeton University Press, NJ.

Weyl, H. (1952). *Symmetry.* Princeton University Press, NJ.

White, D. and Sofge, D. (1992). *Handbook of Intelligent Control.* Van Nostrand Reinhold, London.

Wiener, N. (1948). *Cybernetics.* MIT Press, USA, second edition.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2).

Zhang, H. and Paul, R. (1991). A parallel inverse kinematic solution for robot manipulators based on multiprocessing and linear extrapolation. *IEEE Transactions on Robotics and Automation*, 7(5):660–669.

Zhang, H. and Roth, Z. S. (1993). A linear solution to kinematic parameter identification of robot manipulators. *IEEE Transactions on Robotics and Automation*, 9(2):174–185.

Zhuang, H., Wang, L., and Roth, Z. (1993). Simultaneous calibration of a robot and a hand-mounted camera. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 149–154.